ATME COLLEGE OF ENGINEERING

13th KM Stone, Bannur Road, Mysore - 560 028

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(ACADEMIC YEAR 2024-25)

LESSON NOTES

SUBJECT: USER INTERFACE DESIGN

SUB CODE: 21IS733

SEMESTER: VII

INSTITUTIONAL MISSION AND VISION

Objectives

- To provide quality education and groom top-notch professionals, entrepreneurs and leaders for different fields of engineering, technology and management.
- To open a Training-R & D-Design-Consultancy cell in each department, gradually introduce doctoral and postdoctoral programs, encourage basic & applied research in areas of social relevance, and develop the institute as a center of excellence.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To cultivate strong community relationships and involve the students and the staff in local community service.
- To constantly enhance the value of the educational inputs with the participation of students, faculty, parents and industry.

Vision

- Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

Mission

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torch bearers of tomorrow's society.
- To strive to attain ever-higher benchmarks of educational excellence.

Department of Computer Science & Engineering

Vision of the Department

- develop highly talented individuals in Computer Science and Engineering to deal with real world challenges in industry, education, research and society.

Mission of the Department

- To inculcate professional behaviour, strong ethical values, innovative research capabilities and leadership abilities in the young minds & to provide a teaching environment that emphasizes depth, originality and critical thinking.
- Motivate students to put their thoughts and ideas adoptable by industry or to pursue higher studies leading to research

Program Educational Objectives (PEO'S):

1. Empower students with a strong basis in the mathematical, scientific and engineering fundamentals to solve computational problems and to prepare them for employment, higher learning and R&D.
2. Gain technical knowledge, skills and awareness of current technologies of computer science engineering and to develop an ability to design and provide novel engineering solutions for software/hardware problems through entrepreneurial skills.
3. Exposure to emerging technologies and work in teams on interdisciplinary projects with effective communication skills and leadership qualities.
4. Ability to function ethically and responsibly in a rapidly changing environment by applying innovative ideas in the latest technology, to become effective professionals in Computer Science to bear a life-long career in related areas.

Program Specific Outcomes (PSOs)

1. Demonstrate understanding of the principles and working of the hardware and software aspects of Embedded Systems.
2. Use professional Engineering practices, strategies and tactics for the development, implementation and maintenance of software.
3. Provide effective and efficient real time solutions using acquired knowledge in various domains.

USER INTERFACE DESIGN
SEMESTER – VII

Subject Code	21IS733	CIE Marks	50
Number of Contact Hours/Week	3:0:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total	100
Credits	03	Exam Hours	03

Course Learning Objectives:

CLO 1. To study the concept of menus, windows, interfaces.

CLO 2. To study about business functions.

CLO 3. To study the characteristics and components of windows and the various controls for the windows.

CLO 4. To study about various problems in windows design with color, text, graphics and

CLO 5. To study the testing methods..

Module –1

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.

Module –2

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.

Module –3

System menus and navigation schemes- Structures of menus, Functions of menus, Contents of menus, Formatting of menus, Phrasing the menu**,** Selecting menu choices, Navigating menus, Kinds of graphical menus.

Module–4

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

Module–5

Screen based controls- Operable control, Text control, Selection control, Custom control, Presentation control, Windows Tests-prototypes, kinds of tests.

Course Outcomes: At the end of the course the student will be able to:

CO 1. Understand importance and characteristics of user interface design

CO 2. Apply user interface design process on business functions

CO 3. Demonstrate system menus, navigation schemes and windows characteristics

CO 4. Analyze screen based controls and device based controls

CO 5. Design the prototypes and test plans of user interface

Textbooks:

1. **Wilbert O, Galitz, "The Essential Guide to User Interface Design", John Wiley & Sons, Second Edition 2002**

Reference Books:

1. **Ben Sheiderman, "Design the User Interface", Pearson Education, 1998**
2. **Alan Cooper, " The Essential of User Interface Design", Wiley-Dream Tech Ltd.,2002**

MODULE 1
The Importance of the User Interface

In these times of metaphors, mice, widgets/controls, links, applets, and usability, the user interface is being scrutinized, studied, written about, and talked about like never before. This welcome attention, along with the proliferation of usability laboratories and product testing, has significantly raised the usability of products we are present- ing to our users today. People's voices have finally been heard above the din. Their combined voices, frustrated, fed up with complicated procedures and incomprehensi- ble screens, have finally become overwhelming. "We're no longer going to peacefully accept products that mess up our lives and put everything we work on at risk," they aresaying. They're also saying "That's just the way it is" is no longer tolerable as an an- swer to a problem. Examples of good design, when they have occurred, have been pre- sented as vivid proof that good design is possible.

We developers have listened. Greatly improved technology in the late twentieth century eliminated a host of barriers to good interface design and unleashed a variety of new display and interaction techniques wrapped into a package called the graphicaluser interface, or, as it is commonly called, GUI or "gooey." Almost every graphical platform now provides a style guide to assist in product design. Software to aid the GUI design process proliferates. Hard on the heels of GUIs has come the amazingly fast intrusion of the World Wide Web into the everyday lives of people. Web site design has greatly expanded the range of users and introduced additional interface techniques such as multimedia. (To be fair, in some aspects it has dragged interface design back- wards as well, but more about that later.)

It is said that the amount of programming code devoted to the user interface now ex-ceeds 50 percent. Looking backwards, we have made great strides in interface design. Looking around today, however, too many instances of poor design still abound. Look-ing ahead, it seems that much still remains to be done.

Defining the User Interface

User interface design is a subset of a field of study called *human-computer interaction* (HCI). Human-computer interaction is the study, planning, and design of how people and computers work together so that a person's needs are satisfied in the most effectiveway. HCI designers must consider a variety of factors: what people want and expect, what physical limitations and abilities people possess, how their perceptual and infor- mation processing systems work, and what people find enjoyable and attractive. Tech- nical characteristics and limitations of the computer hardware and software must alsobe considered.

The *user interface* is the part of a computer and its software that people can see, hear,touch, talk to, or otherwise understand or direct. The user interface has essentially twocomponents: input and output. *Input* is how a person communicates his or her needs or desires to the computer. Some common input components are the keyboard, mouse, trackball, one's finger (for touch-sensitive screens), and one's voice (for spoken in- structions). *Output* is how the computer conveys the results of its computations and re- quirements to the user. Today, the most common computer output mechanism is the display screen, followed by mechanisms that take advantage of a person's auditory ca-pabilities: voice and sound. The use of the human senses of smell and touch output ininterface design still remain largely unexplored.

Proper interface design will provide a mix of well-designed input and output mech- anisms that satisfy the user's needs, capabilities, and limitations in the most effective way possible. The best interface is one that it not noticed, one that permits the user to focus on the information and task at hand, not the mechanisms used to present the in-formation and perform the task.

The Importance of Good Design

With today's technology and tools, and our motivation to create really effective and us- able interfaces and screens, why do we continue to produce systems that are inefficientand confusing or, at worst, just plain unusable? Is it because:

1. We don't care?
2. We don't possess common sense?
3. We don't have the time?
4. We still don't know what really makes good design?

I take the view that the root causes are Number 4, with a good deal of Number 3 thrown in. We *do* care. But we never seem to have time to find out what makes good de-sign, nor to properly apply it. After all, many of us have other things to do in addition to designing interfaces and screens. So we take our best shot given the workload and time constraints imposed upon us. The result, too often, is woefully inadequate.

I discounted the "we don't possess common sense" alternative years ago. If, as I have heard thousands of times, interface and screen design were really a matter of com- mon sense, we developers would have been producing *almost identical* screens for sim-

ilar applications and functions for many years. When was the last time you saw two de- signers create almost identical screen solutions, based on the same requirements, with- out the aid of design guidelines or standards (or with them as well)?

A well-designed interface and screen is terribly important to our users. It is their window to view the capabilities of the system. To many, *it is* the system, being one of the few visible components of the product we developers create. It is also the vehicle through which many critical tasks are presented. These tasks often have a direct impact on an organization's relations with its customers, and its profitability.

A screen's layout and appearance affect a person in a variety of ways. If they are con- fusing 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. I've heard of one user who relieved his frustrations with his computer with a couple of well-aimed bullets from a gun. I recently heard of another who, in a moment of ex- treme exasperation and anger, dropped his PC out of his upper-floor office window.

The Benefits of Good Design

Imagine the productivity benefits we could gain through proper design. Based on an actual system requiring processing of 4.8 million screens per year and illustrated in Table 1.1, an analysis established that if poor clarity forced screen users to spend one extra second per screen, almost one additional person-year would be required to process all screens. Twenty extra seconds in screen usage time adds an additional 14 person-years.

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. Separate items, which had been combined on the same display line to conserve space, were placed on separate lines instead. The result: screen users were about 20 percent more productive with the less-crowded ver-sion. Other researchers reformatted a series of screens following many of the same concepts to be described in this book. 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 sav-

Table 1.1 Impact of Inefficient Screen Design on Processing Time

ADDITIONAL SECONDS REQUIRED PER SCREEN IN SECONDS	ADDITIONAL PERSON-YEARS REQUIRED TO PROCESS 4.8 MILLION SCREENS PER YEAR
1	.7
5	3.6
10	7.1
20	14.2

ings of 79 person-years in the affected system. In a second study comparing 500 screens, it was found that the time to extract information from displays of airline or lodging information was 128 percent faster for the best format than for the worst.

Other studies have also shown that the proper formatting of information on screensdoes have

a significant positive effect on performance. Cope and Uliano (1995) found that *one* graphical window redesigned to be more effective would save a company about $20,000 during its first year of use.

In recent years, the productivity benefits of well-designed Web pages have also been scrutinized. Baca and Cassidy (1999) redesigned an organization's home page because users were complaining they were unable to find information they needed. These de- signers established a usability objective specifying that after redesign users should beable to locate the desired information 80 percent of the time. After one redesign, 73 per- cent of the searches were completed with an average completion time of 113 seconds. Ad- ditional redesigns eventually improved the success rate to 84 percent, and reduced the average completion time to 57 seconds. The improvement in search success rate between the first redesign and final redesign was 15 percent; the improvement in search time was about 50 percent. (This study also points out the value of iterative testing and redesign.) Fath and Henneman (1999) evaluated four Web sites commonly used for onlineshopping. Participants performed shopping tasks at each site. In three of the Web sites only about one-half of the shopping tasks could be completed, in the fourth 84 percentwere successful. (In the former, one-third of the shopping tasks could not be completedat all.) The more successful, and more usable, site task completion rate was about 65 percent higher than that of the less successful sites. We can only speculate how this
might translate into dollars.

Other benefits also accrue from good design (Karat, 1997). Training costs are low- ered because training time is reduced, support line costs are lowered because fewer as- sist calls are necessary, and employee satisfaction is increased because aggravation and frustration are reduced. Another benefit is, ultimately, that an organization's cus-tomers benefit because of the improved service they receive.

Identifying and resolving problems during the design and development process also has significant economic benefits. Pressman (1992) has shown that for every dollar spent fixing a problem during product design, $10 would be spent if the problem was fixed during development, and $100 would be spent fixing it after the product's release. A general rule of thumb: every dollar invested in usability returns $10 to $100 (IBM, 2001). How many screens are used each day in our technological world? How many screens are used each day in your

organization? Thousands? Millions? Imagine the possible savings. Proper screen design might also, of course, lower the costs of replac-

ing "broken" PCs.

A Brief History of the Human-Computer Interface

The need for people to communicate with each other has existed since we first walkedupon this planet. The lowest and most common level of communication modes we share are movements and gestures. Movements and gestures are language-

independent, that is, they permit people who do not speak the same language to deal with one another.

The next higher level, in terms of universality and complexity, is spoken language. Most people can speak one language, some two or more. A spoken language is a veryefficient mode of communication if both parties to the communication understand it.

At the third and highest level of complexity is written language. While most peoplespeak, not all can write. But for those who can, writing is still nowhere near as efficient a means of communication as speaking.

In modern times, we have the typewriter, another step upward in communication complexity. Significantly fewer people type than write. (While a practiced typist can find typing faster and more efficient than handwriting, the unskilled may not find this the case.) Spoken language, however, is still more efficient than typing, regardless of typing skill level.

Through its first few decades, a computer's ability to deal with human communica-tion was inversely related to what was easy for people to do. The computer demandedrigid, typed input through a keyboard; people responded slowly using this device and with varying degrees of skill. The human-computer dialog reflected the computer's preferences, consisting of one style or a combination of styles using keyboards, com- monly referred to as Command Language, Question and Answer, Menu Selection, Function Key Selection, and Form Fill-In. For more details on the screens associated with these dialogs see Galitz (1992).

Throughout the computer's history, designers have been developing, with varyingdegrees of success, other human-computer interaction methods that utilize more gen-eral, widespread, and

easier-to-learn capabilities: voice and handwriting. Systems that recognize human speech and handwriting now exist, although they still lack the uni- versality and richness of typed input.

Introduction of the Graphical User Interface

Finally, in the 1970s, another dialog alternative surfaced. Research at Xerox's Palo AltoResearch Center provided an alternative to the typewriter, an interface using a form ofhuman gesturing, the most basic of all human communication methods. The Xerox sys- tems, Altus and STAR, introduced the mouse and pointing and selecting as the primary human-computer communication method. The user simply pointed at the screen, using the mouse as an intermediary. These systems also introduced the graphical user interface as we know it today. Ivan Sutherland at the Massachusetts Institute of Tech- nology (MIT) is given credit for first introducing graphics with his Sketchpad program in 1963. Lines, circles, and points could be drawn on a screen using a light pen. Xerox worked on developing handheld pointing devices in the 1960s and patented a mouse with wheels in 1970. In 1974, Xerox patented today's ball mouse, after a researcher wassuddenly inspired to turn a track ball upside down.

Xerox was never able to market the STAR successfully, but Apple quickly picked up the concept and the Macintosh, released in 1984, was the first successful mass-market system. A new concept was born, revolutionizing the human-computer interface. A chronological history of GUIs is found in Table 1.2.

Table 1.2 Chronological History of Graphical User Interfaces

1973	Pioneered at the Xerox Palo Alto Research Center.
—First to pull together all the elements of the modern GUI.	
1981	First commercial marketing as the Xerox STAR.
—Widely introduced pointing, selection, and mouse.	
1983	Apple introduces the Lisa.

— Features pull-down menus and menu bars.

1984 Apple introduces the Macintosh.

— Macintosh is the first successful mass-marketed system.

1985 Microsoft Windows 1.0 released.

Commodore introduces the Amiga 1000.

1987 X Window System becomes widely available.

IBM's System Application Architecture released.

Including Common User Access (CUA).

IBM's Presentation Manager released.

— Intended as graphics operating system replacement for DOS. Apple introduces the Macintosh II.

— The first color Macintosh.

1988 NeXT's NeXTStep released.

— First to simulate three-dimensional screen.

1989 UNIX-based GUIs released.

— Open Look by AT&T and Sun Microsystems.

— Innovative appearance to avoid legal challenges.

— Motif, for the Open Software Foundation by DEC andHewlett-Packard.

— Appearance and behavior based on Presentation Manager. Microsoft Windows 3.0 released.

1992	OS/2 Workplace Shell released. Microsoft Windows 3.1 released.
1993	Microsoft Windows NT released.
1995	Microsoft Windows 95 released.
1996	IBM releases OS/2 Warp 4. Microsoft introduces NT 4.0.
1997	Apple releases the Mac OS 8.
1998	Microsoft introduces Windows 98.
1999	Apple releases Mac OS X Server.

	— A UNIX-based OS.
2000	Microsoft Windows 2000 released.Microsoft Windows ME released
2001	Microsoft Windows XP released

Characteristics of Graphical and Web User Interfaces

The graphical user interface differed significantly from its text-based forefather. The Web interface differs from a GUI interface in significant ways also, not all differences, how- ever, can be considered interface advancements. In this chapter, the characteristics of a GUI interface will be reviewed, including the concept it introduced: direct manipulation.Then, Web characteristics will be reviewed, including the differences between GUI and Web interface design, and the differences between printed page and Web design.

The Graphical User Interface

In brief, a graphical user interface can be defined as follows. A *user interface,* as recentlydescribed, is a collection of techniques and mechanisms to interact with something. Ina *graphical* interface, the primary interaction mechanism is a pointing device of some kind. This device is the electronic equivalent to the human hand. What the user inter- acts 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 toperform tasks. They are interacted with as entities independent of all other objects. Peo- ple perform operations, called *actions*, on objects. The operations include accessing andmodifying objects by pointing, selecting, and manipulating. All objects have standardresulting behaviors.

The Popularity of Graphics

Graphics revolutionized design and the user interface. A graphical screen bore scant resemblance to its earlier text-based colleagues. Whereas the older text-based screen possessed a one-dimensional, text-oriented, form-like quality, graphic screens assumed a three-dimensional look. Information floated in windows, small rectangular boxes seemed to rise above the background plane. Windows could also float above other win- dows. Controls appeared to rise above the screen and move when activated. Lines ap- peared to be etched into the screen. Information could appear, and disappear, as needed, and in some cases text could be replaced by graphical images called icons. These icons could represent objects or actions.

Screen navigation and commands are executed through menu bars and pull-downs. Menus "pop up" on the screen. In the screen body, selection fields such as radio but- tons, check boxes, list boxes, and palettes coexisted with the reliable old text entry field. More sophisticated text entry fields with attached or drop-down menus of alter-natives also became available. Screen objects and actions were selected through useof pointing mechanisms, such as the mouse or joystick, instead of the traditional keyboard.

Increased computer power and the vast improvement in the display enable the user's actions to be reacted to quickly, dynamically, and meaningfully. This new inter-face is characterized as representing one's "desktop" with scattered notes, papers, and objects such as files, trays, and trashcans arrayed around the screen. It is sometimes re- ferred to as the WIMP interface: windows, icons, menus, and pointers.

Graphic presentation of information utilizes a person's information-processing ca- pabilities much more effectively than other presentation methods. Properly used, it re- duces the requirement for perceptual and mental information recoding and reorganization, and also reduces the memory loads. It permits faster information trans-fer between computers and people

by permitting more visual comparisons of amounts, trends, or relationships; more compact representation of information; and simplifica- tion of the perception of structure. Graphics also can add appeal or charm to the inter-face and permit greater customization to create a unique corporate or organization style.

The Concept of Direct Manipulation

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

The system is portrayed as an extension of the real world. It is assumed that a per-son is already familiar with the objects and actions in his or her environment of in- terest. The system simply replicates them and portrays them on a different medium, the screen. A person has the power to access and modify these objects, among which are windows. A person is allowed to work in a familiar environ- ment 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. Like one's desktop, objects are con- tinuously visible. Reminders of actions to be performed are also obvious, labeled buttons replacing complex syntax and command names. Cursor action and mo- tion occurs in physically obvious and intuitively natural ways. Nelson (1980) de- scribed 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). Rutkowski (1982) described it as "transparency," where one's intellect is applied to the task, not the tool. Hutchins, Hollan, and Norman (1986) considered it direct involvement with the world of objects rather than com- municating with an intermediary.

One problem in direct manipulation, however, is that there is no direct anal- ogy on the desk for all necessary windowing operations. A piece of paper on one's desk maintains a constant size, never shrinking or growing. Windows can do both. Solving this problem required

embedding a control panel, a familiar concept to most people, in a window's border. This control panel is manipulated,not the window itself.

Actions are rapid and incremental with visible display of results. Since tactile feedback is not yet possible (as would occur with one's hand when one touches something), 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 con-tinuous and effortless.

Incremental actions are easily reversible. Finally, actions, if discovered to be incor- rect or not desired, can be easily undone.

Earlier Direct Manipulation Systems

Using the above definition, the concept of direct manipulation actually preceded the first graphical system. The earliest full-screen text editors possessed similar character-istics. Screens of text resembling a piece of paper on one's desk could be created (ex- tension of real world) and then reviewed in their entirety (continuous visibility). Editing or restructuring could be easily accomplished (through rapid incremental ac- tions) and the results immediately seen. Actions could be reversed when necessary. Ittook the advent of graphical systems to crystallize the direct manipulation concept, however.

Indirect Manipulation

In practice, direct manipulation of *all* screen objects and actions may not be feasible be-cause 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 windowborder may be limited.

It may be difficult for people to learn and remember all the necessary operationsand actions.

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

Which style of interaction—direct manipulation, indirect manipulation, or a combi-nation of both—is best, under what conditions and for whom, remains a question whose answer still eludes us.

Graphical Systems: Advantages and Disadvantages

Graphical systems burst upon the office with great promise. The simplified interface they presented was thought to reduce the memory requirements imposed on the user, make more effective use of one's information-processing capabilities, and dramatically reduce system learning requirements. Experience indicates that for many people they have done all these things.

Advantages

The success of graphical systems has been attributed to a host of factors. The followinghave been commonly referenced in literature and endorsed by their advocates as ad- vantages of these systems.

Symbols recognized faster than text. Research has found that symbols can be rec- ognized faster and more accurately than text, and that the graphical attributes oficons, such as shape and color, are very useful for quickly classifying objects, ele-ments, or text by some common property. An example of a good classification scheme that speeds up recognition are the icons developed for indicating the kind of message being presented to the user of the system. The text of an informational message is preceded by an "i" in a circle, a warning message by an exclamation point, and a critical message by another unique symbol. These icons allow speedy recognition of the type of message being presented.

Faster learning. Research has also found that a graphical, pictorial representation aids learning, and symbols can also be easily learned.

Faster use and problem solving. Visual or spatial representation of information hasbeen found to be easier to retain and manipulate and leads to faster and more suc- cessful problem solving. Symbols have also been found to be effective in convey-ing simple instructions.

Easier remembering. Because of greater simplicity, it is easier for casual users to re- tain operational concepts.

More natural. Graphic representations of objects are thought to be more natural and closer to innate human capabilities. In human beings, actions and visual skills emerged before languages. It has also been suggested that symbolic displays are more natural and advantageous because the human mind has a powerful image memory.

Exploits visual/spatial cues. Spatial relationships are usually found to be under- stood more quickly than verbal representations. Visually thinking is believed to be better than logical thinking.

Fosters more concrete thinking. Displayed objects are directly in the high-level task domain, or directly usable in their presented form. There is no need mentally to decompose tasks into multiple commands with complex syntactic form. The need for abstract thinking is therefore minimized.

Provides context. Displayed objects are visible, providing a picture of the current context.

Fewer errors. More concrete thinking affords fewer opportunities for errors. Re- versibility of actions reduces error rates because it is always possible to undo thelast step. Error messages are less frequently needed.

Increased feeling of control. The user initiates actions and feels in control. This in- creases user confidence and hastens system mastery.

Immediate feedback. The results of actions furthering user goals can be seen im- mediately. Learning is quickened. 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. The user has more control. This ability to reverse un- wanted actions also increases user confidence and hastens system mastery.

Less anxiety concerning use. Hesitant or new users feel less anxiety when using thesystem because

it is so easily comprehended, is easy to control, and has pre- dictable responses and reversible actions.

More attractive. Direct-manipulation systems are more entertaining, cleverer, and more appealing. This is especially important for the cautious or skeptical user.

May consume less space. Icons may take up less space than the equivalent in words. More information can often be packed in a given area of the screen. This, how- ever, is not always the case.

Replaces national languages. Language-based systems are seldom universally ap- plicable. Language translations frequently cause problems in a text-based sys- tem. 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 re-verse is not true.

Low typing requirements. Pointing and selection controls, such as the mouse or trackball, eliminate the need for typing skills.

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

Disadvantages

The body of positive research, hypotheses, and comment concerning graphical systems is being challenged by some studies, findings, and opinions that indicate that graphi- cal representation and interaction may not necessarily always be better. Indeed, in some cases, it may be poorer than pure textual or alphanumeric displays. Trying to force all system components into a graphical format may be doing a disservice to the user. Some also feel that, as graphical systems are becoming increasingly sophisticated and continue to expand, interfaces have become increasingly more complex, some- times arcane, and even bizarre. Among the disadvantages put forth are these:

Greater design complexity. The elements and techniques available to the graphicalscreen designer far outnumber those that were at the disposal of the text-based screen designer. Controls and basic alternatives must be chosen from a pile of choices numbering in excess of 50. (Conversely,

alternatives available to the text- based screen designer numbered about 15.) This design potential may not neces- sarily result in better design, unless the choices are thoughtfully selected and consistently and simply applied. Proper window types must also be chosen and colors selected from a seemingly unending rainbow of alternatives. With graph- ics, the skill of the designer is increasingly challenged. Poor design can undermineacceptance.

Learning still necessary. The first time one encounters many graphical systems, what to do is not immediately obvious. The meanings of many words and icons may not be known. It is not often possible to guess their meanings, especially the more ar-bitrary symbols. How to use a pointing device may also have to be learned. A se- vere learning and remembering requirement is imposed on many users, and it takes a while to get up to speed. A text-based system could easily be structured toincorporate a set of clear instructions: (1) Do this, (2) now do this, and so on.

System providers estimate that becoming accustomed to a graphical interfaceshould require about eight hours of training. Other experts say the learning timeis closer to 20 or 30 hours.

Lack of experimentally-derived design guidelines. The graphical interface is still burdened today by a lack of widely available experimentally-derived design guidelines. Early on, more developer interest existed in solving technical rather than usability issues, so few studies to aid in making design decisions were per- formed. Today, studies being performed in usability laboratories are rarely pub- lished. This occurs because of a number of factors. First, builders of platforms and packages will not publish their study results because they want to maintain a competitive advantage. If a better way is found to do something, or present some-thing, why tell the competition? Let them make the same mistake, or find the an-swer themselves.

Second, the studies are often specific to a particular function or task. They may not be generally applicable. Third, it takes time and effort to publish some- thing. The developer in today's office seldom has the time. Finally, it is also diffi- cult to develop studies evaluating design alternatives because of increased GUI complexity. Too many variables that must be controlled make meaningful cause-and-effect relationships very difficult to uncover.

Consequently, there is too little understanding of how most design aspects re- late to productivity and satisfaction.

Inconsistencies in technique and terminology. Many differences in technique, ter-minology, and

look and feel exist among various graphical system providers, and even among successive versions of the same system. These inconsistencies occur because of copyright and legal implications, product differentiation considera- tions, and our expanding knowledge about the interface. The result is that learn-ing, and relearning, for both designers and users is much more difficult than it should be.

Working domain is the present. While direct-manipulation systems provide con- text, they also require the user to work in the "present." Hulteen (1988), in a par- ody of "WYSIWYG," suggests "What you see is all you get." Walker (1989) argued that language takes you out of the here and now and the visually present.Language, she continues, makes it easier to find things.

Not always familiar. Symbolic representations may not be as familiar as words or numbers. We have been exposed to words and numbers for a long time. Research has found that numeric symbols elicit faster responses than graphic symbols in avisual search task. One developer had to modify a new system during testing byreplacing iconic representations with a textual outline format. The users, lawyers,were unfamiliar with icons and demanded a more familiar format.

Human comprehension limitations. Human limitations may also exist in terms of one's ability to deal with the increased complexity of the graphical interface. Thevariety of visual displays can still challenge all but the most sophisticated users. The number of different icons that can be introduced is also restricted because oflimitations in human comprehension. Studies continually find that the number ofdifferent symbols a person can differentiate and deal with is much more limited than text. Some researchers note that claims for the easy understanding of pic- tograms are exaggerated, and that recognizing icons requires much perceptual learning, abstracting ability, and intelligence.

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

Window manipulation requirements. Window handling and manipulation times are still excessive and repetitive. This wastes time and interrupts the decision- making needed to perform tasks and jobs.

Production limitations. The number of symbols that can be clearly produced using today's technology is still limited. A body of recognizable symbols must be pro- duced that are equally

legible and equally recognizable using differing technolo-gies. This is extremely difficult today.

Few tested icons exist. Icons, like typefaces, must appear in different sizes, weights,and styles. As with text, an entire font of clearly recognizable symbols must be de- veloped. It is not simply a question of developing an icon and simply enlarging orreducing it. Changing an icon's size can differentially affect symbol line widths, open areas, and so forth, dramatically affecting its recognizability. Typeface de- sign is literally the product of 300 years of experimentation and study. Icons must be researched, designed, tested, and then introduced into the marketplace. The consequences of poor or improper design will be confusion and lower productiv- ity for users.

Inefficient for touch typists. For an experienced touch typist, the keyboard is a very fast and powerful device. Moving a mouse or some other pointing mechanism may be slower.

Inefficient for expert users. Inefficiencies develop when there are more objects and actions than can fit on the screen. Concatenation for a command language is im- possible.

Not always the preferred style of interaction. Not all users prefer a pure iconic in-terface. A study comparing commands illustrated by icons, icons with text, or text-only, found that users preferred alternatives with textual captions.

Not always fastest style of interaction. Another study has found that graphic in- structions on an automated bank teller machine wereinferior to textual instructions.

Increased chances of clutter and confusion. A graphical system does not guaranteeelimination of clutter on a screen. Instead, the chance for clutter is increased, thereby increasing the possibility of confusion. How much screen clutter one candeal with is open to speculation. The possibility that clutter may exist is evidencedby the fact that many people, when working with a window, expand it to fill theentire display screen. This may be done to reduce visual screen clutter. Mori and Hayashi (1993) found that visible windows, not the focus of attention, degraded performance in the window being worked on.

The futz and fiddle factor. With the proliferation of computer games, computer usage can be wasteful of time. Stromoski (1993) estimates that five hours a weekin the office are spent playing and tinkering. Experts have said that the most usedprogram in Microsoft Windows is Solitaire!

Tinkering includes activities such as creating garish documents reflecting almost every object property (font size, style,color, and so on.) available.

Futzing and fiddling does have some benefits, however. It is a tool for learn- ing how to use a mouse, for example, and it is a vehicle for exploring the system and becoming familiar with its capabilities. It is of value when done in moderation.

May consume more screen space. Not all applications will consume less screen space. A listing of names and telephone numbers in a textual format will be moreefficient to scan than a card file.

Hardware limitations. Good design also requires hardware of adequate power, pro-cessing speed, screen resolution, and graphic capability. Insufficiencies in these areas can prevent a graphic system's full potential from being realized.

Characteristics of the Graphical User Interface

A graphical system possesses a set of defining concepts. Included are sophisticated vi- sual presentation, pick-and-click interaction, a restricted set of interface options, visu- alization, object orientation, extensive use of a person's recognition memory, and concurrent performance of functions.

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, in- cluding different sizes and styles. The display of 16 million or more colors is possible

on some screens. Graphics also permit animation and the presentation of photographs and motion video.

The meaningful interface elements visually presented to the user in a graphical sys- tem 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 thescreen the real world of the user as

realistically, meaningfully, simply, and clearly as possible.

Pick-and-Click Interaction

Elements of a graphical screen upon which some action is to be performed must first beidentified. The motor activity required of a person to identify this element for a pro- posed 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. The user moves the mouse pointer to the relevant element (pick) and the action is signaled (click). Pointing allows rapid selection and feedback. The eye, hand, and mind seem to work smoothly and efficiently together.

The secondary mechanism for performing these selection actions is the keyboard.

Most systems permit pick-and-click to be performed using the keyboard as well.

Restricted Set of Interface Options

The array of alternatives available to the user is what is presented on the screen or whatmay be retrieved through what is presented on the screen, nothing less, nothing more. This concept fostered the acronym WYSIWYG.

Visualization

Visualization is a cognitive process that allows people to understand information thatis difficult to perceive, because it is either too voluminous or too abstract. It involves changing an entity's representation to reveal gradually the structure and/or function of the underlying system or process. Presenting specialized graphic portrayals facili- tates visualization. The best visualization method for an activity depends on what peo- ple are trying to learn from the data. The goal is not necessarily to reproduce a realistic graphical image, but to produce one that conveys the most relevant information. Ef- fective visualizations can facilitate mental insights, increase productivity, and foster faster and more accurate use of data.

Object Orientation

A graphical system consists of objects and actions. *Objects* are what people see on the screen. They are manipulated as a single unit. A well-designed system keeps users fo-cused on objects,

not on how to carry out actions. Objects can be composed of *subobjects.*

For example, an object may be a document. The document's subobjects may be a para- graph, sentence, word, and letter.

IBM's System Application Architecture Common User Access Advanced Interface Design Reference (SAA CUA) (IBM, 1991) breaks objects into three meaningful classes: data, container, and device. *Data objects* present information. This information, either text or graphics, normally appears in the body of the screen. It is, essentially, the screen-based controls for information collection or presentation organized on the screen.

Container objects are objects to hold other objects. They are used to group two or more related objects for easy access and retrieval. There are three kinds of container objects: the workplace, folders, and workareas. The *workplace* is the desktop, the storage area for all objects. *Folders* are general-purpose containers for long-term storage of objects. *Workareas* are temporary storage folders used for storing multiple objects currently being worked on.

Device objects represent physical objects in the real world, such as printers or trash baskets. These objects may contain others for acting upon. A file, for example, may beplaced in a printer for printing of its contents.

Microsoft Windows specifies the characteristics of objects depending upon the rela-tionships that exist between them. Objects can exist within the context of other objects,and one object may affect the way another object appears or behaves. These relation- ships are called collections, constraints, composites, and containers.

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

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

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

A *container* is an object in which other objects exist. Examples include text in a doc- ument or documents in a folder. A container often influences the behavior of its con- tent. It may add or

suppress certain properties or operations of objects placed within it, control access to its content, or control access to kinds of objects it will accept.

These relationships help define an object's *type*. Similar traits and behaviors exist inobjects of the same object type.

Another important object characteristic is *persistence*. Persistence is the maintenanceof a state once it is established. An object's state (for example, window size, cursor lo- cation, scroll position, and so on) should always be automatically preserved when theuser changes it.

Properties or Attributes of Objects

Objects also have properties or attributes. Properties are the unique characteristics of an object. Properties help to describe an object and can be changed by users. Examples of properties are text styles (such as normal or italics), font sizes (such as 10 or 12 points),or window background colors (such as black or blue).

Actions

In addition to objects are actions. People take actions on objects. They manipulate ob- jects in specific ways (commands) or modify the properties of objects (property or at- tribute specification).

Commands are actions that manipulate objects. They may be performed in a variety of ways, including by direct manipulation or through a command button. They are ex- ecuted immediately when selected. Once executed, they cease to be relevant. Examples of commands are opening a document, printing a document, closing a window, and quitting an application.

Property/attribute specification actions establish or modify the attributes or propertiesof objects. When selected, they remain in effect until deselected. Examples include se- lecting cascaded windows to be displayed, a particular font style, or a particular color.

The following is a typical *property/attribute specification sequence:*

1. The user selects an object—for example, several words of text.
2. The user then selects an action to apply to that object, such as the action BOLD.
3. The selected words are made bold and will remain bold until selected andchanged again.

A series of actions may be performed on a selected object. Performing a series of ac-tions on an object also permits and encourages system learning through exploration.

Application versus Object or Data Orientation

Earlier graphical systems were usually application-oriented, a continuation of the phi- losophy that enveloped text-based systems. When a text-based system was developed, it was called an application. As graphical systems evolved, developers usually thought in terms of applications as well. When a real picture of the user began to emerge, it fi-nally became evident that people thought in terms of tasks, not applications. They choose objects and then act upon them.

An application-oriented approach takes an action:object approach, like this:

Action> 1. An application is opened (for example, word processing).Object> 2. A file or other object selected (for example, a memo).

An object-oriented object:action approach does this:

Object> 1. An object is chosen (a memo).

Action> 2. An application is selected (word processing).

The object-action approach permits people to more easily focus on their task and minimizes the visibility of the operating system and separate applications. Many ex- perienced users may have difficulty in switching from one approach to another since an old interaction behavior must be unlearned and a new one learned. New users should not experience these problems, since it more accurately reflects a person's think- ing. In any one interface, it is critical that a consistent orientation be maintained, eitheran object:action or an action:object approach.

Views

Views are ways of looking at an object's information. IBM's SAA CUA describes four kinds of views: composed, contents, settings, and help.

Composed views present information and the objects contained within an object. They are typically associated with data objects and are specific to tasks and products being worked with. *Contents* views list the components of objects. *Settings* views permit see-ing and changing object properties. *Help* views provide all the help functions.

Use of Recognition Memory

Continuous visibility of objects and actions encourages use of a person's more power- ful recognition memory. The "out of sight, out of mind" problem is eliminated.

Concurrent Performance of Functions

Graphic systems may do two or more things at one time. Multiple programs may run simultaneously. When a system is not busy on a primary task, it may process back- ground tasks (cooperative multitasking). When applications are running as truly sep- arate tasks, the system may divide the processing power into time slices and allocate portions to each application (preemptive multitasking). Data may also be transferred between programs. It may be temporarily stored on a "clipboard" for later transfer orbe automatically swapped between programs.

The Web User Interface

The expansion of the World Wide Web since the early 1990s has been truly amazing. Once simply a communication medium for scientists and researchers, its many and per- vasive tentacles have spread deeply into businesses, organizations, and homes around the world. Unlike earlier text-based and GUI systems that were developed and nurtured in an organization's Data Processing and Information Systems groups, the Web's roots were sown in a market-driven society thirsting for convenience and information.

Web interface design is essentially the design of navigation and the presentation of information. It is about content, not data. Proper interface design is largely a matter ofproperly 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 en-vironment. It is also a graphically rich

environment.

Web interface design is difficult for a number of reasons. First, its underlying designlanguage, HTML, was never intended for creating screens to be used by the general population. Its scope of users was expected to be technical. HTML was limited in ob- jects and interaction styles and did not provide a means for presenting information in the most effective way for people. Next, browser navigation retreated to the pre-GUI era. This era was characterized by a "command" field whose contents had to be learned, and a navigational organization and structure that lay hidden beneath a mostly dark and blank screen. GUIs eliminated the absolute necessity for a command field, providing menus related to the task and the current contextual situation. Browser navigation is mostly confined to a "Back" and "Forward" concept, but "back-to-where"and "forward-to-where" is often unremembered or unknown. Ill-timed use of the Back button can destroy many minutes worth of work. Remaining navigation was willed to Web pages themselves, where the situation only worsened. Numerous links were pro- vided to destinations unknown, invisible navigation buttons lay unrecognizable on the screen, and linked jumps two paragraphs down the page were indistinguishable from those that went to the Ukraine. Also, form completion and submission was es- sentially a form of batch processing. Forms were completed, transmitted, and then edited instead of the editing being interactive, occurring as the entry process was ac- complished. Web interface design is now struggling to recover from these giant steps backward.

Web interface design is also more difficult because the main issues concern infor-

mation 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.

Today, then, the Web interface is a victim of its poor foundation. It is also a victim of its explosive and haphazard growth. Looking forward, interface design tools will ma- ture, research-based design guidelines will become increasingly available (and will beapplied), and knowledge of users and their needs will expand. Then, the ultimate goal of a Web that feels natural, is well structured, and is easy to use will reach fruition.

The Popularity of the Web

While the introduction of the graphical user interface revolutionized the user interface, the Web has revolutionized computing. It allows millions of people scattered across the globe to communicate, access information, publish, and be heard. It allows people to control much of the display and the rendering of Web pages. Aspects such as typogra-phy and colors can be changed, graphics turned off, and decisions made whether or not to transmit certain data over nonsecure channels or whether to accept or refuse cook- ies. Nowhere in the history of computing has the user been given so much control.

Web usage has reflected this popularity. The number of Internet hosts has risen dra-matically. In 1984, hosts online exceeded 1,000; in 1987, 10,000; in 1989, 100,000, in 1990, 300,000; in 1992 hosts exceeded one million. Commercialization of the Internet saw even greater expansion of the growth rate. In 1993, Internet traffic was expanding at a 341,634 percent annual growth rate. In 1996, there were nearly 10 million hosts online and 40 million connected people (PBS Timeline).

User control has had some decided disadvantages for some Web site owners as well. Users have become much more discerning about good design. Slow download times, confusing navigation, confusing page organization, disturbing animation, or other un- desirable site features often results in user abandonment of the site for others with a more agreeable interface. People are quick to vote with their mouse, and these warn- ings should not go unheeded.

Characteristics of a Web Interface

A Web interface possesses a number of characteristics, some similar to a GUI interface, and, as has already been shown, some different. In the following paragraphs many of these specific commonalities and differences will be examined. Also, the differing char- acteristics of printed page design and Web page design will be compared.

GUI versus Web Page Design

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

they are composed of many similar components. Significant dif-ferences do exist, however. The following paragraphs highlight the other most signifi- cant differences. Table 2.1 provides a summary listing. Parts of this discussion are based upon Berry (2000) and Nielsen (1997a).

Devices. In *GUI* design, the characteristics of interface devices such as monitors and modems are well defined, and design variations tend to be restricted. Monitor display capabilities, such as installed fonts and screen size, are established and easily considered in the design process. In *Web* design, no assumptions about the user's interface devices can be made. User devices may range from handheld mechanisms to high-end workstations. (In GUI design, the difference in screen area between a laptop and a high-end workstation is a factor of six, in Web page design this difference may be as high as 100.) Connection speed bandwidths mayalso vary by a factor of 1,000. Consequently, WYSIWYG no longer exists in page design. In GUI design, the layout of a screen will look exactly as specified, Web page look will be greatly influenced by both the hardware and software. With the Web, the designer has to relinquish full control and share responsibility for the in-terface with users and their hardware and software.

User focus. *GUI* systems are about well-defined applications and data, about trans- actions and processes. Thorough attention must usually be addressed to tasks in need of completion. The *Web* is about information and navigation, an environ- ment where people move back and forth in an unstructured way among many pages of information. Web use is most often characterized browsing and visual scanning of information to find what information is needed.

Data/information. *GUI* data is typically created and used by known and trusted sources, people in the user's organization or reputable and reliable companies and organizations. The properties of the system's data are generally known, and the information is typically organized in an understandable and meaningful fash- ion. A notion of shared data exists, as does a notion of data privacy. The *Web* is full of unknown content typically placed there by others unknown to the user. Typi-cal users don't put information on the Web (except for publishing their own pages). The reliability and truthfulness of found information cannot always be as- certained and trusted. Web content is usually highly variable in organization, andthe privacy of the information is often suspect.

User tasks. *GUI* system users install, configure, personalize, start, use, and upgrade programs. They open, use, and close data files.

an individual application, and people become familiar with many of its features and its design. *Web* users do things like linking to sites, browsing or reading pages, filling out forms, registering for services, participating in transactions, and downloading and saving things. Movement between pages and sites is often a very rapid activity, with people not gaining familiarity with many sites. The typ- ical Web user has no notion of programs and tends to be much less aware of com- puter mechanics. Most GUI and Web users to not want to spend the effort required to set up or install anything.

User's conceptual space. In a *GUI* environment the user's conceptual space is con- trolled by the program and application. A user's access to data is constrained, and space is made available where their data can be stored and managed. A *Web* user's space is infinite and generally unorganized. Little opportunity for mean- ingful organization of personal information exists.

Presentation elements. The main presentation elements for *GUIs* are various kinds of windows, menus, controls, toolbars, messages, and data. Many elements are transient, dynamically appearing and disappearing based upon the current con- text of the interaction. They are also generally standardized as a result of the toolkits and style guides used. Elements are presented on screens exactly as spec- ified by the designer. *Web* systems possess two components: the browser and page. Many browsers are substantially GUI applications with traditional GUI pre- sentation elements. Within a page itself, however, any combination of text, im- ages, audio, video, and animation may exist. Complex, cluttered, and visually distracting pages are easy to generate and often exist. This occurs because many designers have focused on implementing that which is new, pretty, or attention getting, with little thought given to usability. The availability of interface style guides and guidelines to aid the design process is not known (or they are ig- nored). Common toolkits and industry conventions, however, are now being pro- posed and will be slowly adopted. Also contributing to page design problems is the fact that a page may not be presented exactly as specified by the designer. The exact look of a page is dependent on browser and monitor used. Extreme varia- tions in screen sizes for presenting pages can and do exist. Finally, the user can change the look of a page by modifying its properties.

Navigation. *GUI* users navigate through structured menus, lists, trees, dialogs, and wizards. Paths are constrained by design (grayed out menu choices, for example), and the navigation

mechanisms standardized by toolkits and style guides. Navi- gation is a weakly established concept, a supplement to more important task func-tions and actions. Some aspects of a GUI do provide a strong sense of navigation,the ellipsis on "to another window" intent indicators such as "Open...," commandbuttons such as "OK" and "Cancel" that direct the user's focus to another win- dow, and wizards. Others aspects of GUI design do not provide a strong sense of navigation—button pressing, for example, that does not result in something visi-ble happening (for example, pressing an Apply button).

Web users control their own navigation through links, bookmarks, and typed URLs. Navigation is a significant and highly visible concept with few constraints. The immense size of the Web, and the user's ability to easily wander just about anywhere, frequently causes a lost "sense of place," or "Where am I right now?"

feeling. Web navigation has few standards beyond the browser's Back button and underlined links. Typically most navigation is part of page design that fosters a lack of consistency, and often confuses users. Establishing a continual sense of place for the user is a critical aspect of Web page design.

Context. *GUI* systems enable the user to maintain a better sense of context. Paths arerestricted, and multiple overlapping windows may be presented and be visible, enabling users to remember how what they are doing fits into the overall task pic- ture. *Web* pages are single entities with almost unlimited navigation paths. Theydo not bring up separate dialog boxes to ask questions, provide or request sup- plemental information, or present messages. Contextual clues become limited orare hard to find.

Interaction. *GUI* interactions consist of such activities as clicking menu choices, pressing buttons, selecting choices from list, keying data, and cutting, copying, or pasting within context established by an open window and an active program. The basic *Web* interaction is a single click. This click can cause extreme changes incontext such as moving to another site or changing the displayed information within a site. The user may not notice subtle changes when they occur. Addition- ally, the browser provides parallel mechanisms like the Back button that may function differently depending on context. The distinction between an action and a navigation link is not always obvious.

Response time. Compared to the Web, response times with a *GUI* system are fairlystable, if not

nearly instantaneous. *Web* response times can be quite variable, and often aggravatingly slow. Line transmission speeds, system loads, and page content can have a dramatic impact. Long response times can upset and frustrate users.

Visual style. In *GUI* systems, the visual style is typically prescribed and constrained by toolkit. (Exceptions are entertainment and multimedia applications.) Visual creativity in screen design is allowed but it is difficult to do. While some user op- tions and style choices do exist, little opportunity exists for screen personalization.In *Web* page design, a more artistic, individual, and unrestricted presentation style is allowed and encouraged. Much design freedom exists, but differing browser and display capabilities, multiple screen sizes, and bandwidth limita- tions, often complicate and restrict this freedom. Limited personalization of the system is available, at a browser or site level, for users.

System capability. *GUI* system capabilities are only limited in proportion to the ca-pability of the hardware in terms of speed, memory, and configuration, and the sophistication of the software. The *Web* is more constrained, being limited by con- straints imposed by the hardware, browser, and software. It is also limited by thewillingness of the page owner to provide certain functions and elements, and the willingness of the user to allow features because of response time, security, and privacy issues and concerns.

Task efficiency. *GUI* systems are targeted to a specific audience performing specific tasks. Generally, the efficiency of performing a task is only limited by the amount of programming undertaken to support it. Browser and network capabilities limit *Web* task efficiency. The actual user audience is usually not well understood, sincemany Web sites are intended for anyone and everyone.

Consistency. Consistency in *GUI* system design is a major objective in most devel- opment efforts. While they are far from perfect, an attempt is made to be consis- tent both within applications and across applications. Many organizations possess interface and screen design standards and toolkits to aid in the standard- ization process. Toolkits and guidelines also allow a certain degree of universal consistency in GUI products. In *Web* page design, the heavy emphasis on graph- ics, a lack of design standards, and the desire of Web sites to establish their own identities results in very little consistency across sites. Web sites often establish standards within a site, but in too many instances developers ignore guidelines existing for GUI

components used in Web pages. These problems are especially found in the presentation of screen controls on pages.

User assistance. User assistance is an integral part of most *GUI* systems applications.This assistance is accessed through standard mechanisms such as the F1 key and Help menus. Message and status areas are also provided on the screen. Docu- mentation, both online and offline, is normally provided, as is a support desk to answer user questions and provide guidance and assistance. *Web* pages do not yetprovide similar help systems. What little help that is available is built into the page. Customer service support, if provided, is generally oriented to the productor service offered. GUI browsers may provide GUI-type assistance, so the user sees two different assistance approaches. Deficiencies in Web page help then be-come more obvious.

Integration. A primary goal of most *GUI* applications is the seamless integration of all pieces. Common functions are supported across applications and import/ex- port capabilities exist. Again, toolkits and their components are key elements in accomplishing this objective. In *Web* design, some integration is apparent within a site for basic functions such as navigation and printing. But because sites strive for individual distinction, interoperability between sites is almost nonexistent.

Security. In a *GUI* environment, security and data access can be tightly controlled, inproportion to the degree of willingness of an organization to invest resources andeffort. For home applications, security is not an issue for most PC users. The *Web* is renowned for security exposures. This is a major inhibitor of Web use for both busi- nesses and consumers. Browser-provided security options have typically not been well understood by average Web users. When employed, these security options often have function-limiting side effects. (Disabled cookies, for example.) Attempts to create a more trustworthy appearance are being made through the use of secu- rity levels and passwords to assure users that the Web is a secure environment.

Reliability. Like security, reliability in *GUI* systems is established and controlled inproportion to the degree of willingness of an organization to invest resources and effort. The computer being used influences reliability as does, if applicable, the local area network. Both are in the control of the using organization. *Web* reliabil- ity is susceptible to disruptions from many directions. Telephone line and cable providers, Internet service providers, hosting servers, and remotely

accessed sites all can contribute to the problem. Accessed applications and user mistakes may also cause reliability problems. A lack of reliability can be a great inhibitor of Web use.

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

In developing a Web system, always evaluate each GUI guideline for direct applic-ability in any development effort. Also, do not simply transport a GUI application or design "en toto" to the Web without evaluating it in terms of the implications de- scribed above. Some applications or designs may require significant changes, others a simple "fine-tuning." One so far unmentioned aspect that both GUI and Web systems *do* have in common, is "know your user." Involving them throughout the redesign process will ensure the best transition to the Web. (More about knowing your users fol-lows in Step 1 "Know Your User or Client.")

Printed Pages versus Web Pages

While Internet history spans but a few years, that of the printed page measures more than five and one-half centuries. Research and experience with printed pages throughthese centuries has created a fundamental and accepted set of guidelines for editorial style, element presentation, and text organization. Many of the basic guidelines, clear, comprehensive, and consistent, can and are being applied to Web page design. Web page design, however, is different in many aspects from the design of books, docu- ments, newspapers, and other similar materials. These differences require a rethinking, researching, and reformulating of a number of these guidelines for use in Web page de-sign. Many of these differences have already been identified. Others will surface as Web experience grows and research is conducted. In the following paragraphs, the major differences between print and Web page design are briefly described. Implica- tions for Web page design are also summarized.

Page size. Printed pages are generally larger than their Web counterparts. They are also fixed in size, not variable like Web pages. A printed page can be designed as one entity, the designer being assured that the completed final product will pos- sess an integrated and complete look.

Web pages, while usually designed as a complete entity, are presented in pieces, pieces whose dimensions differ depend- ing on the user's technology (browser, monitor, and so on). The visual impact of the printed page is maintained in hard-copy form, while on the Web all that usu- ally exists are snapshots of page areas. The visual impact of a Web page is sub- stantially degraded, and the user may never see some parts of the page because their existence is not known or require scrolling to bring into view. The design im-plications: the top of a Web page is its most important element, and signals to theuser 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 ele- ments are often rendered slowly, depending upon things like line transmission speeds and page content. Dozens of seconds may be consumed waiting for a page

to completely appear. Impatient users may not wait, moving on to somewhere else. Design implications: Provide page content that downloads fast, and give people elements to read immediately so the sense of passing time is diminished. (The ultimate goal: a bandwidth fast enough to download a Web page as fast as one can turn a paper page.)

Page layout. With the printed page, layout is precise with much attention given to it. With Web pages layout is more of an approximation, being negatively influ- enced 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. Today, the resolution of displayed print characters still exceeds that of screen characters, and screen reading is still slower than reading from a docu- ment. Design implication: Provide an easy way to print long Web documents. (The ultimate goal: a screen resolution sharp enough to render type crisply enough so that screen reading speed reaches that of newspaper reading.)

User focus. Printed pages present people with entire sets of information. Estima- tions of effort needed to deal with the document are fairly easily made, size and the nature of the material being strong contributors. Some printed pages may beread sequentially (a novel) and others (a

newspaper) partially and somewhat se- quentially (the sports section first, perhaps?). Others forms of printed material may simply be skimmed (a sales brochure), but this skimming is usually system-atic in some way. Web pages present people with individual snapshots of infor- mation, often with few clues as to structure and sequence, and rarely with a few cues as to length and depth. People also have a sense that the body of Web infor- mation potentially available is almost unlimited, and that information paths can lead everywhere and anywhere. With few content size cues available and a hugeinformation base, a common resulting behavior of Web users is to skim the infor- mation presented, looking for what is most relevant to their task or need. This is done for personal efficiency and so as not to tax one's patience. Design implica- tions: Create easy to scan pages and limit the word count of textual content. Also, provide overviews of information organization schemes, clear descriptions of where links lead, and estimations of sizes of linked pages and materials.

Page navigation. Navigating printed materials is as simple as page turning. It is a motor skilled learned early in life and never thought of as navigation or a design el-ement. Substantial interaction between pages is rare, since the process is essentially sequential. Navigating the Web requires innumerable decisions concerning which of many possible links should be followed. It requires asking oneself questions such as these: What is at the end of this link? Where is it? Will it address my need or solvemy problem? Design implications are similar to the above—provide overviews of information organization schemes and clear descriptions of where links lead.

Sense of place. With paper documents you derive a sense of where you are through a mixture of graphic and editorial organization, and size cues supplied by the de-sign of the document. The document is an object with physical characteristics. Paging through printed material is an orderly process, sequential and under- standable. Electronic documents provide none of these physical cues. All that is visible is a small collection of text, graphics, and links hinting that much else lies*somewhere* underneath. Moving through the Web links can cause radical shifts in location and context. Paging using the browser's Back button steps one back through links visited and may involve passing through different documents. Fixed locations that provide cues to support one's memory concerning the loca- tion of things are nonexistent. All these factors cause a person to easily lose a sense of place and lead to confusion. Design implication: Build cues into Web pages to aid the user in maintaining a sense of place.

Interactivity. Printed page design involves letting the eyes traverse static informa- tion, 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) inconjunction with the eyes. Information relationships, static or dynamic, are ex- pressed chronologically as part of the interaction and user movements. Doing is more memorable and makes a stronger impact than simply seeing. No print prece- dents exist for this style of interaction. A better understanding of this process (aswell as better hardware and software) is needed to enhance interactivity.

Page independence. Because moving between Web pages is so easy, and almost any page in a site can be accessed from anywhere else, pages must be made freestand- ing. Every page is independent, and its topic and contents must be explained with- out assumptions about any previous page seen by the user. Printed pages, being sequential, fairly standardized in organization, and providing a clear sense of place, are not considered independent. Specific types of content (table of contents, author, index, and so on) are easily found in well-established document locations. Design implication: Provide informative headers and footers on each Web page.

In conclusion, many of the basic print guidelines can be applied to Web page design.As shown above, however, printed material design differs from Web page design in many aspects. New guidelines must continue to be developed, implemented, and mod- ified as necessary as technology advances and our understanding of Web interaction increases. For the moment, apply existing guidelines where relevant, and new guide- lines as necessary. Part 2 of this book describes many of these guidelines. What must beavoided are things that made sense in the print world, but do not meets today's needsin Web interface design.

The Merging of Graphical BusinessSystems and the Web

Another strength of the Web lies in its ability to link databases and processing occur- ring on a variety of machines within a company or organization. Within a closed sys- tem, queries against databases can be made, internal communication performed, and information useful for employees can be made available. Current systems can also be implemented with more

traditional GUI interfaces. The graphical business system andthe Web will merge into a common entity. These Web systems are called intranets.

Characteristics of an Intranet versus the Internet

An intranet has many of the same characteristics as the Internet. They differ, however, in some important ways. The following discussion is partly based upon Nielsen (1997b).

Users. The users of intranets, being organization employees, know a lot about the or-ganization, its structure, its products, its jargon, and its culture. Internet sites areused by customers and others who know much less about the organization, and often care less about it. The intranet user's characteristics and needs can be much more specifically defined then can those of the general Internet user.

Tasks. An intranet is used for an organization's everyday activities, including com- plex transactions, queries, and communications. The Internet is mainly used to find information, with a supplementary use being simple transactions.

Type of information. An intranet will contain detailed information needed for or- ganizational functioning. Information will often be added or modified. The Inter- net 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 or- ganization's Internet site. Massive amounts of information and processes seem to be needed to make an organization function. It has been estimated that an intranetsite can be ten to one hundred times larger than its corresponding public site.

Hardware and software. Since intranets exist in a controlled environment, the kinds of computers, monitors, browsers, and other software can be restricted or stan- dardized. The need for cross-platform compatibility is minimized or eliminated, permitting more predictable design. Upgraded communications also permit in- tranets to run from a hundred to a thousand times faster than typical Internet ac- cess can. This allows the use of rich graphics and multimedia, screen elements that contribute to very slow download times for most Internet users.

Design philosophy. Implementation on the intranet of current text-based and GUI applications will present a user model similar to those that have existed in other domains. This will cause a

swing back to more traditional GUI designs—designs that will also incorporate the visual appeal of the Web, but eliminate many of itsuseless, promotional, and distracting features. The resulting GUI hybrids will bericher and much more effective.

Extranets

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

Principles of User Interface Design

An interface must really be just an extension of a person. This means that the system and its software must reflect a person's capabilities and respond to his or her specific needs. It should be useful, accomplishing some business objectives faster and more ef- ficiently than the previously used method or tool did. It must also be easy to learn, for people want to do, not learn to do. Finally, the system must be easy and fun to use, evoking a sense of pleasure and accomplishment not tedium and frustration.

The interface itself should serve as both a connector and a separator: a connector inthat it ties the user to the power of the computer, and a separator in that it minimizesthe possibility of the participants damaging one another. While the damage the user in-flicts on the computer tends to be physical (a frustrated pounding of the keyboard), the damage caused by the computer is more psychological (a threat to one's self-esteem). Throughout the history of the human-computer interface, various researchers and writers have attempted to define a set of general principles of interface design. What fol- lows is a compilation of these principles. They reflect not only what we know today, but also what we think we know today. Many are based on research, others on the collective thinking of behaviorists working with user interfaces. These principles will continue to evolve, expand, and be refined as our experience with GUIs and the Web increases. We will begin with the first set of published principles, those for the Xerox STAR.

Principles for the Xerox STAR

The design of the Xerox STAR was guided by a set of principles that evolved over its lengthy development process (Smith, Harslem, Irby, Kimball, and Verplank, 1982; Ver- plank, 1988). These principles established the foundation for graphical interfaces.

The illusion of manipulable objects. Displayed objects that are selectable and ma-nipulable must be created. A design challenge is to invent a set of displayable ob- jects that are represented meaningfully and appropriately for the intended application. It must be clear that these objects can be selected, and how to select them must be self-evident. When they are selected should also be obvious, be- cause it should be clear that the selected object will be the focus of the next action. Verplank calls this "graphics with handles on it." Standalone icons easily fulfilled this requirement. The handles for windows were placed in the borders (window-specific commands, pop-up menus, scroll bars, and so on).

Visual order and viewer focus. Attention must be drawn, at the proper time, to theimportant and relevant elements of the display. Effective visual contrast between various components of the screen is used to achieve this goal (STAR was mono- chromatic so color was not used). Animation is also used to draw attention, as is sound. Feedback must also be provided to the user. Since the pointer is usually the focus of viewer attention, it is a useful mechanism for providing this feedback(by changing shapes).

Revealed structure. The distance between one's intention and the effect must be minimized. Most often, the distance between intention and effect is lengthened as system power increases. The relationship between intention and effect must be tightened and made as apparent as possible to the user. The underlying structureis often revealed during the selection process.

Consistency. Consistency aids learning. Consistency is provided in such areas as el-ement location, grammar, font shapes, styles, and sizes, selection indicators, and contrast and emphasis techniques.

Appropriate effect or emotional impact. The interface must provide the appropri- ate 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 de-vice on which it

will be displayed. Quality of screen images will be greatly af- fected by a device's resolution and color-generation capabilities.

General Principles

The design goals in creating a user interface are described below. They are fundamen-tal 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. Specific guidelines on how to implement many of these goals will be presentedin Part 2. The compilation is presented alphabetically, and the ordering is not intendedto imply degree of importance. They are derived from the various principles described in Galitz (1992), IBM (1991, 2001), Mayhew (1992), Microsoft (1992, 1995, 2001), Open

Software Foundation (1993), and Verplank (1988).

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.

A design aesthetic, or visually pleasing composition, is attractive to the eye. It draws attention subliminally, conveying a message clearly and quickly. Visual appeal makesa computer system accessible and inviting. A lack of visually pleasing composition is disorienting, obscures the intent and meaning, and slows down and confuses the user. Visual appeal is terribly important today because most human-computer communica-tion occurs in the visual realm.

Visual appeal is provided by following the presentation and graphic design princi-ples to be discussed, including providing meaningful contrast between screen elements, creating spatial groupings, aligning screen elements, providing three-dimensional rep- resentation, and using color and graphics effectively. Good design combines power, functionality, and simplicity with a pleasing appearance.

Clarity

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

The interface must be clear in visual appearance, concept, and wording. Visual ele- ments should be understandable, relating to the user's real-world concepts and func- tions. Metaphors, or analogies, should be realistic and simple. Interface words and text should be simple, unambiguous, and free of computer jargon.

Compatibility

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

User compatibility. Design must be appropriate and compatible with the needs of the user or client. Effective design starts with understanding the user's needs andadopting the user's point of view. One very common error among designers is to assume that users are all alike. A glance around the office should quickly putthis assumption to rest. Another common error is to assume that all users think, feel, and behave exactly like the developer. Studies have proven otherwise. Users have quite different needs, aspirations, and attitudes than developers. A systemreflecting only the knowledge and attitudes of its designers cannot be successful."Know

the user" is *the* fundamental principle in interface design. User com-

patibility can only happen if understanding truly occurs.

Task and job compatibility. The organization of a system should match the tasks aperson must do to perform the job. The structure and flow of functions should permit easy transition between tasks. The user must never be forced to navigate between applications or many screens to complete routine daily tasks.

Product compatibility. The intended user of a new system is often the user of other systems or earlier versions of the new system. Habits, expectations, and a level of knowledge have been established and will be brought to bear when learning the new system. If these habits, expectations, and knowledge cannot be applied to the new system, confusion results and learning requirements are greatly increased. While compatibility across products must always be considered in relation to im- proving interfaces, making new systems compatible with existing systems will take advantage of what users already know and reduce the necessity for new learning.

Comprehensibility

- Asystemshould be easilylearned 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 asensible

order that is easy to recollect and place in context.

A system should be understandable, flowing in a comprehensible and meaningful order. Strong clues to the operation of objects should be presented. The steps to com- plete a task should be obvious. Reading and digesting long explanations should neverbe necessary.

Configurability

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

Easy personalization and customization through configuration and reconfiguration of a system enhances a sense of control, encourages an active role in understanding, and allows for personal preferences and differences in experience levels. It also leads tohigher user satisfaction.

Some people will prefer to personalize a system to better meet their preferences. Other people will not, accepting what is given. Still others will experiment with recon-figuration and then give up, running out of patience or time. For these latter groups of users a good default configuration must be provided.

Consistency

- A system should look, act, and operate the same throughout. Similar componentsshould:
  - 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.

Design consistency is the common thread that runs throughout these guidelines. It is the cardinal rule of all design activities. Consistency is important because it can re- duce requirements for human learning by allowing skills learned in one situation to betransferred to

another like it. While any new system must impose some learning re- quirements on its users, it should avoid encumbering productive learning with non- productive, unnecessary activity. Consistency also aids learning of the system's mental model.

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.

Inconsistencies in design are caused by differences in people. Several designers might each design the same system differently. Inconsistencies also occur when those per- forming design activities are pressured by time constraints. All too often the solutions in those cases are exceptions that the user must learn to handle. People, however, per- ceive a system as a single entity. To them, it should look, act, and feel similar through- out. Excess learning requirements become a barrier to users achieving and maintaining high performance and can ultimately influence user acceptance of the system.

Can consistency make a big difference? One study found that user thinking time nearly doubled when the position of screen elements, such as titles and field captions, was varied on a series of menu screens.

Design consistency is achieved by developing and applying design standards or guidelines. In the late 1980s the computer industry and many using organizations fi- nally awakened to the need for them, and a flurry of graphical user interface guidelinedocuments were developed and published. These guidelines specify the appearance and behavior of the GUI user interface. They describe the windows, menus, and vari- ous controls available, including what they look like and how they work. They also provide some guidance on when to use the various components.

Examples of industry-produced guidelines include Apple's *Macintosh Human Inter- face Guidelines* (1992b), Digital Equipment Corporation's *XUI Style Guide* (1988), IBM's *System Application Architecture Common User Access* (1987, 1989a, 1989b, 1991, 1992), Sun Microsystem's *OPEN LOOK Graphical User Interface Application Style Guidelines*

(1990), Open Software Foundation's *OSF/MOTIF Style Guide* (1993), and Microsoft's *The Windows Interface* (1992) and *The Windows Interface Guidelines for Software Design* (1995).

The Web has burst upon the scene with few standards and guidelines to direct de- sign. Many GUI and printed material principles are applicable but they have been ap- plied in a haphazard manner. New research-based guidelines are desperately needed. Organizations working on traditional interface guidelines or standards include the International Standards Organization (ISO), the American National Standards Institute (ANSI), and the Human Factors and Ergonomics Society (Billingsley, 1996). The devel- opment of Web design guidelines has been one focus of the World Wide Web Consor-tium (2001).

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.

Control is feeling in charge, feeling that the system is responding to your actions. Feeling that

a machine is controlling you is demoralizing and frustrating. The inter- face should present a tool-like appearance. Control is achieved when a person, work- ing at his or her own pace, is able to determine what to do, how to do it, and then is easily able to get it done. Simple, predictable, consistent, flexible, customizable, and passive interfaces provide control. Lack of control is signaled by unavailable systems, long delays in system responses, surprising system actions, tedious and long pro- cedures that cannot be circumvented, difficulties in obtaining necessary informa- tion, and the inability to achieve the desired results. The feeling of control has been found to be an excellent mitigator of the work stress associated with many automated systems

In general, avoid modes since they restrict the actions available to the user at any given time. If modes must be used, they should be visually obvious (for example, a changed mouse pointer shape). Existing modes must also be easy to learn and easy toremove.

Directness

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

Tasks should be performed directly. Available alternatives should be visible, reduc- ing the user's mental workload. Directness is also best provided by the object-action sequence of direct-manipulation systems. Tasks are performed by directly selecting an object, then selecting an action to be performed, and then seeing the action being performed.

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.

Eye and hand movements must not be wasted. One's attention must be captured byrelevant elements of the screen when needed. Sequential eye movements between screen elements should be predictable, obvious, and short. Web pages must be easily scannable. All navigation paths should be as short as possible. Manual transitions be- tween various system controls should also be as short as possible. Avoid frequent tran- sitions between input devices such as the keyboard and mouse.

Always try to anticipate the user's wants and needs. At each step in a process, pre- sent to the user all the information and tools needed to complete the process. Do not re-quire user to search for and gather necessary information and tools.

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.

Build on the user's existing knowledge. Build into the interface concepts, terminol- ogy, workflows, and spatial arrangements already familiar to the user. Operations should mimic one's behavior patterns; dialogs should mimic one's thought processes and vocabulary. Familiar concepts enable people to get started and become productivequickly.

Flexibility

- A system must be sensitive to the differing needs of its users, enabling a level andtype 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.

Flexibility is the system's ability to respond to individual differences in people. Per- mit people to choose the method of interaction that is most appropriate to their situa- tion. People should be able to interact with a system in terms of their own particular needs, including knowledge, experience, and personal preference. Flexibility is accom- plished by providing multiple ways to access application functions and perform tasks.It is also accomplished through permitting system customization. Another benefit of flexibility is that it contributes to increased user control. A flexible system is a versatilesystem.

Flexibility is not without dangers. Highly flexible systems can confuse inexperienced people, causing them to make more errors. For this reason, flexibility appears desirable only for experienced users. The novice user should not be exposed to system flexibility at the start, but only as experience is gained. The concept of "progressive disclosure," to be discussed in the simplicity guideline to follow, is also applicable here.

Another problem with flexibility is that it may not always be used, some people pre- fer to continue doing things in the way they first learned. A variety of factors may ac-count for this, including an unwillingness to invest in additional learning, or, perhaps, new ways may not be obvious. The former problem may be addressed by making the new ways as easy and safe to learn as possible, the latter by including in training and reference materials not only information about how to do things, but when they are likely to be useful.

Forgiveness

- Tolerate and forgive common and unavoidable human errors.
- Prevent errors from occurring whenever possible.
- Protect against possible catastrophic errors.

- When an error does occur, provide constructive messages.

It is often said “to err is human.” The corollary to that statement, at least in computersystems, might be “to forgive is good design.” People will make mistakes; a system should tolerate those that are common and unavoidable. A forgiving system keeps people out of trouble.

People like to explore and learn by trial and error. A system oversensitive to erro- neous inputs will discourage users from exploring and trying new things. Learning will be inhibited, and people will be overcautious, working slowly and carefully to avoid mistakes. Productivity will then suffer. A fear of making a mistake and not being able to recover from it has always been a primary contributor to fear of dealing with computers.

Prevent errors from occurring by anticipating where mistakes may occur and de- signing to prevent them. Permit people to review, change, and undo actions whenever necessary. Make it very difficult to perform actions that can have tragic results. Whenerrors do occur, present clear instructions on how to correct them.

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.

Tasks, displays, and movement through a system should be anticipatable based onthe user’s previous knowledge or experience. All actions should lead to results the user expects. Screen elements should be distinct and recognizable. Current operationsshould provide clues as to what will come next. Anticipation, or predictability, reduces mistakes and enables tasks to be completed more quickly. All expectations possessed by the user should be fulfilled uniformly and completely. Predictability is greatly en- hanced by design consistency.

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.

A person should be able to retract an action by issuing an *undo* command. Knowing that an action can be reversed reduces much of the distress of new users, who often worry about doing something wrong. The return point could be the previous action, previous screen, a recent closure point, or the beginning of some predetermined pe- riod, such as back 10 screens or some number of minutes. Reversing or abolishing an action is analogous to using an eraser to eliminate a pencil mark on a piece of paper.

The goal is stability, or returning easily to the right track when a wrong track has been taken. Recovery should be obvious, automatic, and easy and natural to perform. In short, it should be hard to get into deep water or go too far astray. Easy recovery from an actiongreatly facilitates learning by trial and error and exploration. If an action is not reversible, and its consequences are critical, it should be made difficult to accomplish. Always en- sure that users never lose their work as a result of their own errors or technical glitches.

Responsiveness

- The system must rapidly respond to the user's requests.
- Provide immediate acknowledgment for all user actions:

— Visual.

— Textual.

— Auditory.

A user request must be responded to quickly. Knowledge of results, or feedback, is a necessary learning ingredient. It shapes human performance and instills confidence.All requests to the system must be acknowledged in some way. Feedback may be vi- sual, the change in the

shape of the mouse pointer, or textual, taking the form of a mes- sage. It may also be auditory, consisting of a unique sound or tone.

Never leave the screen blank for more than a moment, because the user may think the system has failed. If a request requires an unusually long processing time, or one that is longer than customary, provide an interim "in-progress" message. Also provide some unique form of communication if a user action results in a problem or possible problem.

Substantial or more informative feedback is most important for the casual or new system user. Expert users are often content to receive more modest feedback.

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 madeharder.
  - Provide uniformity and consistency.

Simplicity is the opposite of complexity. Complexity is a measure of the number of choices available at any point in the human-computer interaction. A great deal of func- tionality and power is usually associated with high complexity. Complexity most often

overwhelms and confuses new and casual users of systems. Complex systems are often not fully used, or used ineffectively, because a person may follow known but more cumbersome methods instead of easier but undiscovered or unfamiliar methods.

A system lacking complexity may have a different set of faults. It may be tedious touse or not accomplish much. It is better, however, to provide less functionality that will get effectively used

than to provide too much functionality, yielding an interface hope-lessly complex and extremely difficult to use. Complexity, then, is a two-edged sword.To effectively solve problems, it must be present without being apparent. The goal, then, is to provide a complex system while masking the complexity through a simple interface. There are several ways to minimize this complexity.

Progressive disclosure. Introduce system components gradually so the full com- plexity of the system is not visible at first encounter. Teach fundamentals first. Then, slowly introduce advanced or more sophisticated functions. This is called the layered, or spiral, approach to learning. Such an approach was first describedby Carroll and Carrithers (1984) who called it the "Training-Wheels System." They found that by disabling portions of the system that were not needed and could lead to errors and confusion, improved system learning efficiency was achieved.

Provide defaults. Providing defaults is another form of system layering. When a system is first presented, provide a set of defaults for all system-configurable items. The new user will not be burdened with these decisions and can concen- trate on the fundamentals first. Defaults can later be changed, if desired, as expe-rience increases.

Minimize screen alignment points. A larger number of alignment points of ele- ments displayed on a screen are associated with greater screen visual complexity. Minimizing these alignment points minimizes visual complexity. This concept will be discussed more fully later.

Make common actions simple. Make common actions within a system easier to accomplish than uncommon actions. The benefit will be greater overall system efficiency.

Provide uniformity and consistency. Inconsistency is really a foolish form of com-plexity. It forces a person to learn that things that appear different are not different.

Transparency

- Permit the user to focus on the task or job, without concern for the mechanics of theinterface.
- Workings and reminders of workings inside the computer should be invisible tothe user.

Never force the user to think about the technical details of the system. One's thoughts must be directed to the task, not the computer communication process. Re- minders of the mechanics of the interface occur through the use of technical jargon, the heavy use of codes, and the

presentation of computer concepts and representations.

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.

Design guidelines often cover a great deal of territory and often conflict with one an-other or with technical requirements. In such conflicts the designer must weigh the al- ternatives and reach a decision based on trade-offs concerning accuracy, time, cost, andease of use. Making these trade-offs intelligently requires a thorough understanding of the user and all design considerations. The ultimate solution will be a blend of experi-mental data, good judgment, and the important user needs.

This leads to a second cardinal rule of graphical system development: *Human re- quirements always take precedence over technical requirements.* It may be easier for the de- signer to write a program or build a device that neglects user ease, but final system judgment will always come down to one simple fact: How well does the system meet the needs of the user?

MODULE II

USER INTERFACE DESIGN PROCESS

It is or-ganized in the order of the development steps typically followed in creating a graphi-cal system's or Web site's screens and pages. In total, 14 steps are presented, beginningwith "Know Your User or Client" and ending with a discussion of testing. Other top- ics addressed include considerations in screen design, navigation, screen-based con- trols, writing messages, and text, color, and graphics. This organization scheme enables all the interface design activities to be addressed easily, clearly, and sequentially.

Let us first look at several critical general aspects of the design the process. "Obsta-cles and Pitfalls in the Development Path" points out the realities of designing for peo- ple, and some reasons why design may not live up to expectations. "Designing for People: The Five Commandments" lists the guidelines that are the cornerstones of the entire design process. Then, the concept of *usability*, the primary objective on any de- velopment effort, is defined and discussed. Finally, the desired composition of the in- terface development *design team* is described.

Obstacles and Pitfalls in the Development Path

Developing a computer system is never easy. The path is littered with obstacles and traps, many of them human in nature. Gould (1988) has made these general observa- tions about design:

- Nobody ever gets it right the first time.Development is chock-full of surprises.
- Good design requires living in a sea of changes.
- Making contracts to ignore change will never eliminate the need for change.
- Even if you have made the best system humanly possible, people will still make mistakes when using it.
- Designers need good tools.
- You must have 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. Guidelines in the various design steps address this problem. Behavioral design goals are reviewed in Step 2 "Understand the Business Function."

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.

"Know Your User or Client" is addressed in Step 1, prototypes and testing are ad-dressed in Step 14 "Test, Test, and Retest." The development team is discussed shortly.

Designing for People: The Five Commandments

The complexity of a graphical or Web interface will always magnify any problems thatdo 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 cus- tomers. 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. A wide gap in technical abilities, goals, and at-titudes often exists between users and developers. A failure to understand the dif-ferences will doom a product or system to failure.

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 re- sistance 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. Involvement in design removes the unknown and gives the user a stake in the system or identification with it. One caution, however: user involvement should be based on job or task knowledge, not status or position. The boss seldom knows what is really happening out in the office.

Perform rapid prototyping and testing. Prototyping and testing the product will quickly identify problems and allow you to develop solutions. The design process is complex and human behavior is still not well understood. While the design guidelines that follow go a long way toward achieving ease of use, all problems cannot possibly be predicted. Prototyping and testing must be continually per- formed 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. In many instances, people may adapt to, or become dependent upon, a design, even if it is inefficient. This also makes future modifications much more difficult.

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 developerto revisit a previous stage. This is normal and should be expected. Establish user performance and acceptance criteria and continue testing and modifying until alldesign goals are met.

Integrate the design of *all* the system components. The software, the documenta- tion, the help function, and training needs are all important elements of a graph-ical system or Web site and all should be developed concurrently. A system is being constructed, not simply software. Concurrent development of all pieces will point out possible problems earlier in the design process, allowing them to bemore effectively addressed. Time will also exist for design trade-offs to be thoughtout more carefully.

Usability

Bennett (1979) was the first to use the term *usability* to describe the effectiveness of human performance. In the following years a more formal definition was proposed by Shackel (1981) and modified by Bennett (1984). Finally, Shackel (1991) simply defined usability 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 cycle and should be continually applied throughout the process. The assessment should include the user's entire experience, and all the product's important components. Us- ability assessment methods are discussed more fully in Step 14 "Test, Test, and Retest."

Common Usability Problems

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

1. Ambiguous menus and icons.
2. Languages that permit only single-direction movement through a system.
3. Input and direct manipulation limits.
4. Highlighting and selection limitations.
5. Unclear step sequences.
6. More steps to manage the interface than to perform tasks.
7. Complex linkage between and within applications.
8. Inadequate feedback and confirmation.
9. Lack of system anticipation and intelligence.

10. Inadequate error messages, help, tutorials, and documentation.

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

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

Impaired information readability. Page readability is diminished by poor devel- oper choices in typefaces, colors, and graphics. Use of innumerable typefaces and kaleidoscopic colors wrestle meaning from the screen. A person's attention is di-rected towards trying to understand why the differences exist, instead of being fo- cused toward identifying and understanding the page's content. Backgrounds that are brightly colored or contain pictures or patterns greatly diminish the legi-bility of the overwritten text.

Incomprehensible components. Some design elements give the user no clue as to their function, leaving their purpose not at all obvious. Some icons and graphics, for example, are shrouded in mystery, containing no text to explain what they do. Some buttons don't look at all like command buttons, forcing the user to "mine- sweep" the screen with a mouse to locate the objects that can be used to do something. Command buttons or areas that give no visual indication that they are clickable often won't be clicked. Language is also often confusing, with the de- veloper's terminology being used, not that of the user.

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

Confusing navigation. A site's structure often resembles a maze of twisting pages into which the

user wanders and is quite soon lost. Poor, little, or no organizationexists among pages. The size and depth of many Web sites can eventually lead to a "lost in space" feeling as perceived site structure evaporates as one navigates. Embarking on a side trip can lead to a radical change in context or a path with no signposts or landmarks. Navigation links lead to dead-ends from which there is no return, or boomerang you right back to the spot where you are standing without you being aware of it. Some navigation elements are invisible. (See mystery icons and minesweeping above.) Confusing navigation violates expectations and results in disturbing unexpected behavior.

Inefficient navigation. A person must transverse content-free pages to find what is meaningful. One whole screen is used to point to another. Large graphics waste screen space and add to the page count. The path through the navigation maze isoften long and tedious. Reams of useless data must be sifted through before a need can be fulfilled. Massive use of short pages with little content often creates the feeling that one is "link drunk."

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

Excessive or inefficient page scrolling. Long pages requiring scrolling frequently lead to the user's losing context as related information's spatial proximity increasesand some information entirely disappears from view and, therefore, from mem- ory. Out of sight is often out of mind. If navigation elements and important content are hidden below the page top, they may be missed entirely. To have to scroll to do something important or complete a task can be very annoying; especially if the scrolling is caused by what the user considers is an irrelevancy or noise.

Information overload. Poorly organized or large amounts of information taxes one's memory and can be overwhelming. Heavy mental loads can result from making de- cisions concerning which links to follow and which to abandon, given the large number of choices available. Or from trying to determine what information is im-portant, and what is not. Or from trying to maintain one's place in a huge forest of information trees. One easily becomes buried in decisions and information. Requir-ing even minimal amounts of learning to use a Web site adds to the mental

Design inconsistency. Design inconsistency has not disappeared with the Web. It has been

magnified. The business system user may visit a handful of systems in one day, the Web user may visit dozens, or many more. It is expected that site dif- ferences will and must exist because each Web site owner strives for its own iden- tity. For the user's sake, however, some consistency must exist to permit a seamless flow between sites. Consistency is needed in, for example, navigation el-ement location on a page and the look of navigation buttons (raised). The indus- try is diligently working on this topic and some "common practices" are already in place. The learning principle of rote memorization, however, is still being re- quired *within* many sites. For example, the industry practice of using different standard link colors for unvisited sites (blue) and visited sites (purple) is often vi- olated. This forces users to remember different color meanings in different places, and this also causes confusion between links and underlined text. Design guide-lines for graphical user interfaces have been available for many years. Too often they are ignored (or the designer is unaware of them). Examples of inappropriateuses abound in design. The use of check boxes instead of radio buttons for mutu- ally exclusive options, for example. Or the use of drop-down list boxes instead ofcombination boxes when the task mostly requires keyboard form fill-in. The Web is a form of the graphical user interface, and GUI guidelines should be followed.

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

Stale design caused by emulation of printed documents and past systems. The Web is a new medium with expanded user interaction and information display possibil- ities. While much of what we have learned in the print world and past information systems interface design can be ported to the Web, all of what we know should notbe blindly moved from one to the other. Web sites should be rethought and re- designed using the most appropriate and robust design techniques available.

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

They have existed since designers began build- ing user interfaces.

Some Practical Measures of Usability

Usability, or the lack thereof, can often be sensed by a simple observation of, or talkingto, people using an interface. While these measures lack scientific rigor, they do pro- vide an indication that there may be usability problems.

Are people asking a lot of questions or often reaching for a manual? Many ques- tions or frequent glances at manuals are signs that things are not as clear and in-tuitive as they should be. When in doubt, the first reaction of many people is to ask someone for assistance. When no one is around, then we look in a manual.

Are frequent exasperation responses heard? "Oh damn!" or similar reactions are usually used to express annoyance or frustration. Their frequency, and loudness,may foretell a strong rejection of a product. The absence of exasperation, however,may not represent acceptance. Some people are not as expressive in their lan- guage, or are better able to smother their feelings.

Are there many irrelevant actions being performed? Are people doing things the hard way? Are there incidental actions required for, but not directly related to, doing a job? These include excessive mouse clicks or keyboard strokes to accom- plish something, or going through many operations to find the right page in a manual or the right window or page in the display.

Are there many things to ignore? Are there many elements on the screen that the user must disregard? Are there many "doesn't pertain to me" items? If so, remember, they still consume a portion of a person's visual or information- processing capacities, detracting from the capacities a person could devote to rel-evant things.

Do a number of people want to use the product? None of us goes out of our way tomake our own lives more difficult. (Unfortunately, other people may, however.) We tend to gravitate to things easy to work with or do. If a lot of people want to use it, it probably has a higher usability score. Attitudes may be a very powerfulfactor in a system's or Web site's acceptance.

Some Objective Measures of Usability

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

How *effective* is the interface? Can the required range of tasks be accomplished:

- At better than some required level of performance (for example, in terms of speed and errors)?
- By some required percentage of the specified target range of users? Within some
- required proportion of the range of usage environments?

How *learnable* is the interface? Can the interface be learned:

- Within some specified time from commissioning and start of user training? Based on
- some specified amount of training and user support?
- Within some specified relearning time each time for intermittent users?

How *flexible* is the interface? Is it flexible enough to:

- Allow some specified percentage variation in tasks and/or environments be- yond those first specified?

What are the *attitudes* of the users? Are they:

- Within acceptable levels of human cost in terms of tiredness, discomfort, frus- tration, and personal effort?
- Such that satisfaction causes continued and enhanced usage of the system?

Human performance goals in system use, like any other design goal, should be stated in quantitative and measurable ways. Without performance goals you will never know if you have achieved them, or how successful the system really is. Clear and con- crete goals also provide objectives for usability testing and ensure that a faulty or un- satisfactory product will not be released.

Values for the various criteria should be specified in absolute terms. An absolute goal might be "Task A must be performed by a first-time user in 12 minutes with no er- rors with 30 minutes of training and without referring to a manual." Goals may also be set in relative terms. For example, "Task B must be performed 50 percent faster than itwas using the previous system."

The level of established goals will depend on the capabilities of the user, the capabil-ities of the system, and the objectives of the system. In addition to providing commit- ments to a certain

level of quality, goals become the foundation for the system test plan.

The Design Team

- Provide a balanced design team, including specialists in:Development
- Human factors
- Visual design Usability assessmentDocumentation 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 devel- opment to define requirements and write the software, human factors specialists to de- fine behavioral requirements and apply behavioral considerations, and people with good visual design skills. Also needed are people skilled in testing and usability as- sessment, documentation specialists, and training specialists.

Also, select team members who can effectively work and communicate with one an-other. To optimize communication, locate the team members in close proximity to oneanother.

KNOW YOUR USER OR CLIENT

The journey into the world of interface design and the screen design process must begin with an understanding of the system user, the *most important* part of any com- puter system. It is the user whose needs a system is built to serve. Understanding peo- ple and what they do is a difficult and often undervalued process but very critical because of the gap in knowledge, skills, and attitudes existing between system users and developers that build them. To create a truly usable system, the designer must al-ways 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.

Understanding How People Interactwith Computers

We will start by looking at some characteristics of computer systems, past and present,that have caused, and are causing, people problems. We will then look at the effect these problems have.

Why People Have Trouble with Computers

Although system design and its behavioral implications have come under intense scrutiny in the last decade or so, as we have seen, this has not always been the case. His-torically, the design of business computer systems has been the responsibility of pro- grammers, systems analysts, and system designers, many of whom possess extensive technical knowledge but little behavioral training. In recent years the blossoming of theWeb, with its extensive graphical capabilities, has found graphic artists being added todesign teams. Like those who have come before them, most

graphical artists also pos- sess extensive technical knowledge in their profession but little training in usability. Design decisions, therefore, have rested mostly on the designers' intuition concerning the user's capabilities and the designer's wealth of specialized knowledge. Consequently, poorly designed interfaces have often gone unrecognized.

The intuition of designers or of anyone else, no matter how good or bad they may be at what they do, is error-prone. It is much too shallow a foundation on which to base design decisions. Specialized knowledge lulls one into a false sense of security. It en- ables one to interpret and deal with complex or ambiguous situations on the basis of context cues not visible to users, as well as a knowledge of the computer system that users do not possess. The result is a system that appears perfectly useful to its design-ers but one that the user is unable or unwilling to face up to and master.

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. Systems often speak in a strange language. Words that are completely alien to the office or home environment or used in different contexts, such as *file- spec*, *abend*, *segment*, and *boot*, proliferate. Learning to use a system often requireslearning a new language.

Non-obvious design. Complex or novel design elements are not obvious or intu- itive, but they must nevertheless be mastered. Operations may have prerequisite conditions that must be satisfied before they can be accomplished, or outcomes may not always be immediate, obvious, or visible. The overall framework of the system may be invisible, with the effect that results cannot always be related to the actions that accomplish them.

Fine distinctions. Different actions may accomplish the same thing, depending upon when they are performed, or different things may result from the same ac-tion. Often these distinctions are minute and difficult to keep track of. Critical dis- tinctions are not made at the appropriate time, or distinctions having no real consequence are made instead, as illustrated by the user who insisted that prob- lems were caused by pressing the Enter key "in the wrong way."

Disparity in problem-solving strategies. People learn best by doing. They have trouble following directions and do not always read instructions before taking anaction. Human problem solving can best be characterized as "error-correcting" or "trial and error," whereby a tentative solution

is formulated based on the avail- able evidence and then tried. This tentative solution often has a low chance of suc- cess, but the action's results are used to modify one's next attempt and so increasethe chance of success. Most early computer systems, however, have enforced an

"error-preventing" strategy, which assumes that a person will not take an action until a high degree of confidence exists in its success. The result is that when peo-ple head down wrong one-way paths, they often get entangled in situations diffi-cult, or impossible, to get out of. The last resort action? Turn off the computer andstart again.

Design inconsistency. The same action may have different names: for example, "save" and "keep," "write" and "list." The same command may cause different things to happen. The same result may be described differently: for example, "not legal" and "not valid." Or the same information may be ordered differently on different screens. The result is that system learning becomes an exercise in rote memorization. Meaningful or conceptual learning becomes very difficult.

Responses to Poor Design

Unfortunately, people remember the one thing that went wrong, not the many that goright, so problems are ascribed an abnormal level of importance. Errors are a symptom of problems. The magnitude of errors in a computer-based system has been found to beas high as 46 percent for commands, tasks, or transactions. Errors, and other problemsthat befuddle one, lead to a variety of psychological and physical user responses.

Psychological

Typical psychological responses to poor design are:

Confusion. Detail overwhelms the perceived structure. Meaningful patterns are dif- ficult to ascertain, and the conceptual model or underlying framework cannot be understood or established.

Annoyance. Roadblocks that prevent a task being completed, or a need from being satisfied, promptly and efficiently lead to annoyance. Inconsistencies in design, slow computer reaction times, difficulties in quickly finding information, out- dated information, and visual screen distractions are a few of the many things that may annoy users.

Frustration. An overabundance of annoyances, an inability to easily convey one's in-tentions to the computer, or an inability to finish a task or satisfy a need can cause frustration. Frustration is heightened if an unexpected computer response cannot be undone or if what really took place cannot be determined. Inflexible and un- forgiving systems are a major source of frustration.

Panic or stress. Unexpectedly long delays during times of severe or unusual pres- sure may introduce panic or stress. Some typical causes are unavailable systems or long response times when the user is operating under a deadline or dealing with an irate customer.

Boredom. Boredom results from improper computer pacing (slow response times orlong download times) or overly simplistic jobs.

These psychological responses diminish user effectiveness because they are severe blocks to concentration. Thoughts irrelevant to the task at hand are forced to the user'sattention, and necessary concentration is impossible. The result, in addition to higher error rates, is poor performance, anxiety, and dissatisfaction.

Physical

Psychological responses frequently lead to, or are accompanied by, the following phys- ical reactions.

Abandonment of the system. The system is rejected and other information sources are relied upon. These sources must, of course, be available and the user must have the discretion to perform the rejection. In business systems this is a common reaction of managerial and professional personnel. With the Web, almost all userscan exercise this option.

Partial use of the system. Only a portion of the system's capabilities are used, usu- ally those operations that are easiest to perform or that provide the most benefits.Historically, this has been the most common user reaction to most computer sys- tems. Many aspects of many systems often go unused.

Indirect use of the system. An intermediary is placed between the would-be user and the computer. Again, since this requires high status and discretion, it is an- other typical response of managers or others with authority.

Modification of the task. The task is changed to match the capabilities of the system. This is a

prevalent reaction when the tools are rigid and the problem is unstruc- tured, as in scientific problem solving.

Compensatory activity. Additional actions are performed to compensate for system inadequacies. A common example is the manual reformatting of information to match the structure required by the computer. This is a reaction common to work- ers whose discretion is limited, such as clerical personnel.

Misuse of the system. The rules are bent to shortcut operational difficulties. This re- quires significant knowledge of the system and may affect system integrity.

Direct programming. The system is reprogrammed by its user to meet specific needs. This is a typical response of the sophisticated worker.

These physical responses also greatly diminish user efficiency and effectiveness. They force the user to rely upon other information sources, to fail to use a system's complete capabilities, or to perform time-consuming "work-around" actions.

People and Their Tasks

The user in today's office is usually overworked, fatigued, and continually interrupted.The home user may also experience these same conditions, and often the pressures as- sociated with children and family life as well. All computer users do tend to share thefollowing: they tend not to read documentation, they do not understand well the prob- lems the computer can aid in solving, and they know little about what information is available to meet their needs. Moreover, the users' technical skills have often been greatly overestimated by the system designer, who is usually isolated psychologically and physically from the users' situation. Unlike the users, the designer is capable of re-solving most system problems and ambiguities through application of experience and technical knowledge. Often the designer cannot really believe that anyone is incapableof using the system created.

The user, while being subjected to the everyday pressures of the office and home, frequently does not care about how technically sophisticated a system or Web site is. The user may even be computer illiterate, and possibly even antagonistic. He or she wants to spend time using a computer, not learning to use it. His or her objective is sim- ply to get some work done, a task performed, or a need satisfied. Today, many users have also learned to expect certain level of

design sophistication. It is in this environ- ment our system will be placed.

Important Human Characteristics in Design

We are complex organisms with a variety of attributes that have an important influence on interface and screen design. Of particular importance in design are perception, memory, visual acuity, foveal and peripheral vision, sensory storage, information pro- cessing, learning, skill, and individual differences.

Perception

Perception is our awareness and understanding of the elements and objects of our en-vironment through the physical sensation of our various senses, including sight, sound, smell, and so forth. Perception is influenced, in part, by *experience*. We classify stimuli based on models stored in our memories and in this way achieve understand-ing. In essence, we tend to match objects or sensations perceived to things we already know. Comparing the accumulated knowledge of the child with that of an adult in in- terpreting the world is a vivid example of the role of experience in perception.

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 com- mon 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 phys-ical size.

Succinctness. We see an object as having some perfect or simple shape because per- fection or simplicity is easier to remember.

Closure. Our perception is synthetic; it establishes meaningful wholes. If somethingdoes not quite close itself, such as a circle, square, triangle, or word, we see it as closed anyway.

Unity. Objects that form closed shapes are perceived as a group.

Continuity. Shortened lines may be automatically extended.

Balance. We desire stabilization or equilibrium in our viewing environment. Vertical,horizontal, and right angles are the most visually satisfying and easiest to look at.

Expectancies. Perception is also influenced by expectancies; sometimes we perceive not what is there but what we expect to be there. Missing a spelling mistake in proofreading something we write is often an example of a perceptual expectancyerror; we see not how a word *is* spelled, but how we *expect* to see it spelled.

Context. Context, environment, and surroundings also influence individual percep- tion. 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 peo- ple have said about the size of the lines.

Signals versus noise. Our sensing mechanisms are bombarded by many stimuli, some of which are important and some of which are not. Important stimuli are called signals; those that are not important or unwanted are called noise. Signals are more quickly comprehended if they are easily distinguishable from noise in our sensory environment. Noise interferes with the perception of signals to the ex- tent that they are similar to one another. Noise can even mask a critical signal. For example, imagine a hidden word puzzle where meaningful words are buried in a large block matrix of alphabetic characters. The signals, alphabetic characters con- stituting meaningful words, are masked by the matrix of meaningless letters.

The elements of a screen assume the quality of signal or noise, depending on the actions and thought processes of the user. Once a screen is first presented and has to be identified as being the correct one, the screen's title may be the signal, the other elements it contains simply being noise. When the screen is being used, the data it contains becomes the signal, and the title now reverts to noise. Other el-ements of the screen rise and fall in importance, assuming the roles of either sig- nals or noise, depending on the user's needs of the moment. The goal in design is to allow screen elements to easily assume the quality of signal or noise, as the needs and tasks of the user change from moment to moment.

The goal in design, then, is to utilize our perceptual capabilities so a screen can be structured

in the most meaningful and obvious way.

Memory

Memory is not the most stable of human attributes, as anyone who has forgotten why they walked into a room, or forgotten a very important birthday, can attest. Today, memory is viewed as consisting of two components, long-term and short-term (or working) memory. This has not always been the case. In the 1950s, most researchers be- lieved there was only one memory system; the short-term component was not recog- nized or accepted. It was in this era that the classic memory study was published (Miller, 1956) indicating that memory limit is 7 ± 2 "chunks" of information. Shortly after this the concept of a short-term memory was identified and, in the 1970s, the view of short-term memory was broadened and called "working memory."

Short-term, or working, memory receives information from either the senses or long- term memory, but usually cannot receive both at once, the senses being processed sep-arately. 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. Based upon research over the years, estimates of working memory storage capacity has gradually been low- ered from Miller's 7 ± 2 items to a size of 3–4 items today.

Knowledge, experience, and familiarity govern the size and complexity of the infor- mation that can be remembered. To illustrate, most native English-speaking people would find remembering English words much easier than remembering an equal num- ber of words in Russian. For a Russian-speaking person the opposite would be true. Short-term memory is easily overloaded. It is highly susceptible to the interference of such distracting tasks as thinking, reciting, or listening, which are constantly erasing and overwriting it. Remembering a telephone number long enough to complete the di- aling operation taxes the memory of many people.

In performance, research indicates that a greater working memory is positively re- lated to increased reading comprehension, drawing inferences from text, reasoning skill, and learning technical information (Baddeley, 1992). Research indicates, as well, that when performing

complex tasks, working memory can be increased through applying two senses, vision and audition, rather than one (Williams, 1998). Research also indi- cates that performance can be degraded when a person must attend to multiple infor-mation sources, and then must integrate the information before understanding occurs. *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 learn- ing. It is a complex process requiring some effort on our part. The learning process isimproved 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. The human ac- tive vocabulary (words that can be recalled) typically ranges between 2,000 and 3,000 words. Passive vocabulary (words that can be recognized) typically numbers about 100,000. Our power of recognition, therefore, is much greater than our power of recall, and this phenomenon should be utilized in design. To do this, one should present,
whenever possible, lists of alternatives to remind people of the choices they have.

MAXIM Minimize the need for a mighty memory.

Other general ways to reduce user memory loads, reduce the need for mental inte-gration, and expand working memory, thus enhancing system usability include:

- Presenting information in an organized, structured, familiar, and meaningfulway.
- Placing all required information for task performance in close physical proximity. Giving
- the user control over the pace of information presentation.

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 envi- ronment, 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.

Though seemingly overwhelmed at times by noise, it can occasionally detect, proverbially, a tree through a forest. One good example is what is sometimes called the "cocktail party affect." Have you ever been at a party when, across the room, through the din of voices, someone mentions your name, and you hear it? In spite of the noise,your radar was functioning.

Repeated and excessive stimulation can fatigue the sensory storage mechanism, making it less attentive and unable to distinguish what is important (called *habituation*). Avoid unnecessarily stressing it. Design the interface so that all aspects and elements serve a definite purpose. Eliminating interface noise will ensure that important things will be less likely to be missed.

Visual Acuity

The capacity of the eye to resolve details is called *visual acuity*. It is the phenomenon that results in an object becoming more distinct as we turn our eyes toward it and rapidly losing distinctness as we turn our eyes away—that is, as the visual angle from the point of fixation increases. It has been shown that relative visual acuity is approxi-mately halved at a distance of 2.5 degrees from the point of eye fixation (Bouma, 1970).Therefore, a five-degree diameter circle centered around an eye fixation character on adisplay has been recommended as the area near that character (Tullis, 1983) or the maximum length for a displayed word (Danchak, 1976).

If one assumes that the average viewing distance of a display screen is 19 inches (475mm), the size of the area on the screen of optimum visual acuity is 1.67 inches (41.8 mm) in diameter. Assuming "average" character sizes and character and line spacings,the number of characters on a screen falling within this visual acuity circle is 88, with 15 characters being contained on the widest line, and seven rows being consumed, as illustrated in Figure 1.1.

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 and will have implications for screen grouping guidelines to be presented later. (Remember, it is the physical size of the circle, five de-grees, that is critical, not the number of characters. A larger or smaller character size will decrease or increase the number of viewable characters.)

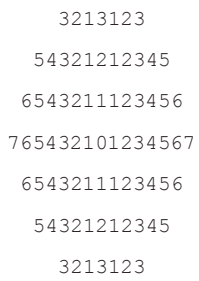

Figure 1.1 Size of area of optimum visual acuity on a screen.

The eye is also never perfectly steady as it sees; it trembles slightly. This tremor im-proves the detection of edges of objects being looked at, thus improving acuity. This tremor, however, can sometimes create problems. Patterns of closely spaced lines or dots are seen to shimmer. This movement can be distracting and disturbing. Patterns for fill-in areas of screens (bars, circles, and so on.) must be carefully chosen to avoid this visual distraction.

Foveal and Peripheral Vision

Foveal vision is used to focus directly on something; *peripheral vision* senses anything inthe area surrounding the location we are looking at, but what is there cannot be clearlyresolved because of the limitations in visual acuity just described. Foveal and periph- eral vision maintain, at the same time, a cooperative and a competitive relationship. Pe- ripheral vision can aid a visual search, but can also be distracting.

In its cooperative nature, peripheral vision is thought to provide clues to where the eye should go next in the visual search of a screen. Patterns, shapes, and alignments pe- ripherally visible can guide the eye in a systematic way through a screen.

In its competitive nature, peripheral vision can compete with foveal vision for atten-tion. What is sensed in the periphery is passed on to our information-processing system along with what is actively being viewed foveally. It is, in a sense, visual noise. Mori and Hayashi (1993) experimentally evaluated the effect of windows in both a foveal and pe-ripheral relationship and found that performance on a foveal window deteriorates whenthere are peripheral windows, and the performance degradation is even greater if the in- formation in the peripheral is dynamic or moving. Care should be exercised in design to utilize peripheral vision in its positive nature,

avoiding its negative aspects.

Information Processing

The information that our senses collect that is deemed important enough to do some- thing about then has to be processed in some meaningful way. Recent thinking (Lind, Johnson, and Sandblad, 1992) is that 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 un- derstanding. You are utilizing this higher level now reading this book.

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 informa- tion rapidly, in parallel with the higher level, and without conscious effort. We look rather than see, perceive rather than read. Repetition and learning results in a shift of control from the higher level to the lower level.

Both levels function simultaneously, the higher level performing reasoning and problem solving, the lower level perceiving the physical form of information sensed. You've probably experienced this difference in working with screens. When a screen is displayed, you usually will want to verify that it is the one you want. If you're new to a system, or if a screen is new to you, you rely on its concrete elements to make that determination, its title, the controls and information it contains, and so forth. You consciously look at the screen and its components using this higher-level processing.

As you become experienced and familiar with screens, however, a newly presented screen can be identified very quickly with just a momentary glance. Just its shape andstructure adequately communicate to you that it is the correct screen for the context in which you are working. Your reasoning and problem solving continues unhindered; your lower-level information processing has assumed the screen identity task.

What assists this lower-level information processing? Visual distinctiveness of a screen is a strong contributor. If a screen is jammed with information and cluttered, it loses its uniqueness and can only be identified through the more time-consuming, andthought-interrupting, reading

process.

Mental Models

As a result of our experiences and culture, we develop mental models of things and peo- ple we interact with. 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 in- teract with another person. Mental models also enable a person to predict the actionsnecessary to do things if the action has been forgotten or has not yet been encountered. When confronting a new computer system, people will bring their own expectations and preconceptions based upon mental models they have formed doing things in their daily life. If the system conforms to the mental models a person has developed, the model is reinforced and the system's use feels more "intuitive." If not, difficulties in learning to use the system will be encountered. This is why in design it is critical that a

user's mental models be to identified and understood.

A person already familiar with one computer system will bring to another system a mental model containing specific visual and usage expectations. If the new system complies with already-established models, it will be much easier to learn and use. The key to forming a transferable mental model of a system is design consistency and de- sign standards.

Movement Control

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

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

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

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

Big buttons are better than small buttons. They provide a larger target for the user to access with the screen pointer. Create toolbar icons that "bleed" into the edges of a dis-play, rather than those leaving a one-pixel non-clickable edge along the display bound- ary. The edge of the screen will stop or "pin" the pointer's movement at a position over toolbar, permitting much faster movement to the toolbar. A one-pixel edge will require more careful positioning of the pointer over the toolbar.

Learning

Learning, as has been said, is the process of encoding in long-term memory informa- tion that is contained in short-term memory. It is a complex process requiring some ef-fort on our part. Our ability to learn is important—it clearly differentiates people frommachines. Given enough time people can improve their performance in almost any task. Too often, however, designers use our learning ability as an excuse to justify com- plex design. Because people can be taught to walk a tightrope is no excuse for incor- porating tightropes in a design when walkways are feasible.

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. Unproductive time spent learning is something frequently avoided.

Regarding the learning process, evidence derived from studies of people learning acomputer system parallels that found in studies of learning in other areas. People pre- fer to be active, to explore, and to use a trial-and-error approach. There is also evidence that people are very sensitive to even minor changes in the user interface, and that suchchanges may lead to problems in transferring from one system to another. Moreover, just the "perception" of having to learn huge amounts of information is enough to keep some people from even using a system. Learning can be enhanced if it:

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

Skill

The goal of human performance is to perform skillfully. To do so requires linking in- puts and responses into a sequence of action. The essence of skill is performance of ac- tions or movements in the correct time sequence with adequate precision. It is characterized by consistency and economy of effort. Economy of effort is achieved by establishing a work pace that represents optimum efficiency. It is accomplished by in- creasing mastery of the system through such things as progressive learning of short- cuts, increased speed, and easier access to information or data.

Skills are hierarchical in nature, and many basic skills may be integrated to form in-creasingly complex ones. Lower-order skills tend to become routine and may drop out of consciousness. System and screen design must permit development of increasinglyskillful performance.

Individual Differences

In reality, there is no average user. A complicating but very advantageous human char-acteristic is that we all differ—in looks, feelings, motor abilities, intellectual abilities, learning abilities and speed, and so on. In a keyboard data entry task, for example, the best typists will probably be twice as fast as the poorest and make 10 times fewer errors. Individual differences complicate design because the design must permit people with widely varying characteristics to satisfactorily and comfortably learn the task or job, or use the Web site. In the past this has usually resulted in bringing designs down to thelevel of lowest abilities or selecting people with the minimum skills necessary to per- form a job. But technology now offers the possibility of tailoring jobs to the specific needs of people with varying and changing learning or skill levels. Multiple versions of a system can easily be created. Design must provide for the needs of all

potential users.

Human Interaction Speeds

The speed at which people can perform using various communication methods has been studied by a number of researchers. The following, as summarized by Bailey (2000), have been found to be typical interaction speeds for various tasks. These speeds are also summarized in Table 1.3.

Reading. The average adult, reading English prose in the United States, has a read-ing speed in the order of 250–300 words per minute. Proofreading text on paper has been found to occur at about 200 words per minute, on a computer monitor, about 180 words per minute (Ziefle, 1998).

Nontraditional reading methods have also been explored in research labora- tories. One technique that has dramatically increased reading speeds is called Rapid Serial Visual Presentation, or RSVP. In this technique single words are pre-sented one at a time in the center of a screen. New words continually replace oldwords at a rate set by the reader. Bailey (1999a) tested this technique with a sam-ple of people whose paper document reading speed was 342 words per minute. (With a speed range of 143 to 540 words per minute.) Single words were pre-sented on a screen in sets at a speed sequentially varying ranging from 600 to 1,600 words per minute. After each set a comprehension test was administered.

Table 1.3 Average Human Interaction Speeds

Reading	
Prose text:	250–300 words per minute.
Proofreading text on paper:	200 words per minute.
Proofreading text on a monitor:	180 words per minute.
Listening:	150–160 words per minute.
Speaking to a computer:	105 words per minute. After recognition

corrections:	25 words per minute.
Keying	
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.

For measured comprehension scores of 75 percent or higher, the average reading speed was 1,212 words per minute. This is about 3.5 times faster than reading in the traditional way. Bailey concludes that computer technology can help improve reading speeds, but nontraditional techniques must be used.

Listening. Words can be comfortably heard and understood at a rate of 150 to 160 words per minute. This is generally the recommended rate for audio books and video narration (Williams, 1998). Omoigui, et al, (1999) did find, however, that when normal speech is speeded up using compression, a speed of 210 words perminute results in no loss of comprehension.

Speaking. Dictating to a computer occurs at a rate of about 105 words per minute (Karat, et al., 1999; Lewis, 1999). Speech recognizer misrecognitions often occur, however, and when word correction times are factored in, the speed drops sig- nificantly, to an average of 25 words per minute. Karat, et al. (1999) also found that the speaking rate of new users was 14 words per minute during transcription and 8 words per minute during composition.

Keying. Fast typewriter typists can key at rates of 150 words per minute and higher.Average typing speed is considered to be about 60–70 words per minute. Com- puter keying has been found to

be much slower, however. Speed for simple tran-scription found by Karat, et al. (1999) was only 33 words per minute and for composition only 19 words per minute. In this study, the fastest typists typed at only 40 words per minute, the slowest at 23 words per minute. Brown (1988) reports that two-finger typists can key memorized text at 37 words per minute andcopied text at 27 words per minute. Something about the computer, its software, and the keyboard does seem to significantly degrade the keying process. (And two-finger typists are not really that bad off after all.)

Hand printing. People hand print memorized text at about 31 words per minute.

Text is copied at about 22 words per minute (Brown, 1988).

Understand the Business Function

A thorough understanding of the user has been obtained, and the focus now shifts to the business function being addressed. Requirements must be determined and user ac-tivities being performed must be described through task analysis. From these, a con- ceptual model of the system will be formulated. Design standards must also be created (if not already available), usability goals established, and training and documentationneeds determined.

A detailed discussion of all of these topics is beyond the scope of this book. The reader in need of more detail is referred to books exclusively addressing systems analy- sis, task analysis, usability, training, and documentation. The general steps to be per- formed are:

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

Business Definition and Requirements Analysis

The objective of this phase is to establish the need for a system. A requirement is an objective that must be met. A product description is developed and refined, based on input from users or marketing. There are many techniques for capturing information for determining requirements. Keil and Carmel (1995), Popowicz (1995), and Fuccella et al. (1999) described many of the methods summarized in Table 2.1 and discussed shortly. They have also provided insights into their advantages and disadvantages. The techniques listed are classified as direct and indirect. Direct methods consist of face-to-face meetings with, or actual viewing of, users to solicit requirements. Indirect methods impose an intermediary, someone or something, between the

users and the developers

Before beginning the analysis, the developer should be aware of the policies and work culture of the organization being studied. He or she should also be familiar with any current system or process the new system is intended to supplement or replace.

Table 2.1 Some Techniques for Determining Requirements

DIRECT METHODS
Individual Face-to-Face Interview • A one-on-one visit with the user to obtain information. It may be structured or somewhat open-ended.
Telephone Interview or Survey • A structured interview conducted via telephone.
Traditional Focus Group • A small group of users and a moderator brought together to verbally discuss therequirements.
Facilitated Team Workshop • A facilitated, structured workshop held with users to obtain requirementsinformation. Similar to the Traditional Focus Group.
Observational Field Study • Users are observed and monitored for an extended time to learn what they do.
Requirements Prototyping • A demo, or very early prototype, is presented to users for comments concerningfunctionality.
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 constructedlaboratory.
Card Sorting for Web Sites • A technique to establish groupings of information for Web sites.

INDIRECT METHODS

MIS Intermediary

- A company representative defines the user's goals and needs to designers anddevelopers.

Paper Survey or Questionnaire

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

Electronic Survey or Questionnaire

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

Electronic Focus Group

- A small group of users and a moderator discuss the requirements online usingworkstations.

Marketing and Sales

- Company representatives who regularly meet customers obtain suggestions orneeds, current and potential.

Support Line

- Information collected by the unit that helps customers with day-to-day problems isanalyzed (Customer Support, Technical Support, Help Desk, etc.).

E-Mail or Bulletin Board

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

User Group

- Improvements are suggested by customer groups who convene periodically to discuss software usage.

Competitor Analyses

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

Trade Show

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

Direct Methods

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

provide the opportunity to immediately follow up on vague or incomplete data. Here are some recommended direct methods for getting input from users.

Individual Face-to-Face Interview

A one-on-one visit is held with the user. It may be structured or more open-ended. Theinterview must have focus and topics to be covered must be carefully planned so data is collected in a common framework, and to ensure that all important aspects are thor-oughly covered. A formal questionnaire should not be used, however. Useful topics to ask the user to describe in an interview include:

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

It is also very useful to also uncover any: Potential measures of system usability

- Unmentioned exceptions to standard policies or procedures.
- Relevant knowledge the user must possess to perform the activity.

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

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

Time must also be allowed for free conversation in interviews. Recording the session for playback to the entire design team provides all involved with some insights into user needs.

Advantages of a personal interview are that you can give the user your full attention,can easily include follow-up questions to gain additional information, will have more time to discuss topics in detail, and will derive a deeper understanding of your users, their experiences, attitudes, beliefs, and desires. Disadvantages of interviews are that they can be costly and time-consuming to conduct, and someone skilled in interview-ing techniques should perform them.

The interviewer must establish a positive relationship with the user, ask questions in a neutral manner, be a good listener, and know when and how to probe for more information.

Telephone Interview or Survey

This interview is conducted using the telephone. It must have structure and be well planned. Arranging the interview in advance allows the user to prepare for it. Tele- phone interviews are less expensive and less invasive than personal interviews. They can be used much more frequently and are extremely effective for very specific infor- mation. Telephone interviews have some disadvantages. It is impossible to gather con- textual 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 (8 to 12) and a moderator are brought together to discuss the re-

quirements. While the discussion is loosely structured, the range of topics must be de-termined beforehand. A typical session lasts about two hours. The purpose of a focus group is to probe user's experiences, attitudes, beliefs, and desires, and to obtain their reactions to ideas or prototypes. Focus groups are not usually useful for establishing how users really work or what kinds of usability problems they really have. Focus group discussion can be influenced by group dynamics, for good or bad. Recording of the session, either video or audio, will permit later detailed analysis of participants comments. Again, the recording can also be played for the entire design team, pro- viding insights into user needs for all developers. Setting up focus group involves thefollowing:

- 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 re- mains 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 team workshop is similar in structure and content to a traditional focus group but is slightly less formal. A common technique used in system requirements de- termination for many years, it is now being replaced (at least in name) by focus groups. Team workshops have had the potential to provide much useful information. Like focus groups, they do require a great deal of time to organize and run.

Observational Field Study

To see and learn what users actually do, they are watched and followed in their own environment, office, or home, in a range of contexts for a period of time. Observation provides good insight into tasks being performed, the working environment and con-ditions, the social

environment, and working practices. It is more objective, natural, and realistic. Observation, however, can be time-consuming and expensive. Video recording of the observation sessions will permit detailed task analysis. Playing the recording for the entire design team again provides all involved with some insights into user tasks.

Requirements Prototyping

A demonstration model, or very early prototype, is presented to users for their com- ments concerning functionality. Prototypes are discussed more fully in Step 14, "Test, Test, and Retest."

User-Interface Prototyping

A demonstration model, or early prototype, is presented to users to uncover user- interface issues and problems. Again, prototypes are discussed more fully in Step 14.

Usability Laboratory Testing

A special laboratory is constructed and users brought in to perform actual newly de- signed tasks. They are observed and results measured, and evaluated to establish the usability of the product at that point in time. Usability tests uncover what people actu- ally do, not what they think they do, a common problem with verbal descriptions. Thesame scenarios can be presented to multiple users, providing comparative data from several users. Problems uncovered may result in modification of the requirements. Us-ability labs can generate much useful information but are expensive to create and op- erate. Usability labs are also discussed in Step 14.

Card Sorting for Web Sites

This is a technique used to establish hierarchical groupings of information for Web sites. It is normally used only after gathering substantial site content information using other analysis techniques. Potential content topics are placed on individual index cards and users are asked to

sort the cards into groupings that are meaningful to them. Card sorting assists in building the site's structure, map, and page content. Briefly, the process is as follows:

- From previous analyses, identify about 50 content topics and inscribe them on index cards. Limit topics to no more than 100.
- Provide blank index cards for names of additional topics the participant may want to add, and 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 timeand 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 par- ticular 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 differ- ent sorting sessions.

The sorting can also be accomplished on the Web. The National Institute of Standards and

Technology (NIST, 2001) has developed a card-sorting tool. The designer sets up the cards and names the categories. The user then sorts by dragging and dropping.

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. Using an intermediary can certainly provide useful information. Working through an intermediary, however, takes away the multichannel communication ad- vantages of face-to-face user-developer contact. Some electronic intermediaries do pro- vide some advantages, as will be described. Imposition of a human intermediary can also create these additional problems. First, there may be a filtering or distortion of themessage, either intentional or unintentional. Next, the intermediary may not possess acomplete, or current, understanding of user's needs, passing on an incomplete or in- correct message. Finally, the intermediary may be a mechanism that discourages direct user-developer contact for political reasons. Indirect methods include the following.

MIS Intermediary

A company representative who defines the user's goals and needs to designers and de- velopers fulfills this intermediary role. This representative may come from the Infor- mation Services department itself, or he or she may be from the using department. While much useful information can be provided, all too often this person does not have the breadth of knowledge needed to satisfy all design requirements.

Paper Survey or Questionnaire

A paper questionnaire or survey is administered to a sample of users to obtain their needs. Questionnaires have the potential to be used for a large target audience locatedmost anywhere, and are much cheaper than customer visits. They generally, however, have a low return rate, often generating responses only from those "very happy" or "very unhappy." They may take a long time to collect and may be difficult to analyze.Questionnaires are useful for determining a user's attitudes, experiences and desires, but not for determining actual tasks and behaviors. Questionnaires should be com- posed mostly of closed questions (yes/no, multiple choice, short

answer, and so on). Open-ended questions require much more analysis. Questionnaires should be rela- tively short and created by someone experienced in their design.

Electronic Survey or Questionnaire

A questionnaire or survey is administered to a sample of users via e-mail or the Web. Characteristics, advantages, and disadvantages are similar to paper surveys and ques- tionnaires. They are, however, significantly less expensive then mailed surveys. The speed of their return can also be much faster than those distributed in a paper format. 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 an- swers 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.

Also 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. Among other things, the detailed survey content can address questions the initial survey raises. A useful follow-up survey goal is to ask the participant to prioritize their needs and to rank expected user tasks according to their importance. A third follow-up survey can also be designed to gather additional information about the most important require- ments and tasks. Iterative surveys, of course, take a longer time to complete. Don't for-get to thank participants for their help and time.

Electronic Focus Group Similar

An electronic focus group is similar to a traditional focus group except that the discus- sion is accomplished electronically using specialized software on a workstation, e-mail,or a Web site. As with the direct methods, the opportunity to immediately follow up onvague or incomplete data exists. All comments, ideas, and suggestions are available inhard-copy form for easier analysis. Specialized software can provide ratings or rank- ings of items presented in lists, a task

requiring much more effort in a traditional focusgroup.

Other advantages of electronic focus groups over traditional focus groups are that the discussion is less influenced by group dynamics; has a smaller chance of being dominated by one or a few participants; can be anonymous, leading to more honest comments and less caution in proposing new ideas; can generate more ideas in a shorter time since all participants can communicate at once; and can lead to longer ses- sions since the participant is in a more comfortable "home environment" and not con- fined to a conference room. Among the disadvantages are that 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. This information is collected inexpensively, since the represen- tative is going to visit the company anyway. Business representatives do have knowl- edge of the nature of customers, the business, and the needs that have to be met. Somedangers: the information may be collected from the wrong people, the representative may unintentionally bias questions, there may be many company "filters" between the representative's contact and the end user, and quantities may sometimes be exagger- ated. ("Lots of people are complaining about . . ." may mean only one or two.) The de- velopers should know the interests and bias of the representatives collecting the information.

Support Line

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

E-Mail, Bulletin Boards or Guest Book

Problems, questions, and suggestions by users posted to a bulletin board, a guest book,or through

e-mail are gathered and evaluated. Again, the focus of this method is usually only on problems. The responsibility is on the user to generate the recommendations, but this population often includes unhappy users. This is a fairly inexpensive method.

User Group

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

Competitor Analysis

Reviews of competitor's products, or Web sites, can also be used to gather ideas, un- cover design requirements, and identify tasks. The designers can perform this evalua- tion or, even better, users can be asked to perform the evaluation.

Trade Show

Customers at a trade show can be exposed to a mock-up or prototype and asked for comments. This method is dependent on the knowledge level of the customers and may provide only a superficial view of most prominent features.

Other Media Analysis

Analyze how other media, print or broadcast, present the process, information, or sub-ject matter of interest. Findings can be used to gather ideas, uncover design require- ments, and identify better ways to accomplish or show something.

System Testing

New requirements and feedback stemming from ongoing system testing can be accu- mulated,

evaluated, and implemented as necessary.

Determining Basic Business Functions

A detailed description of what the product will do is prepared. Major system func- tions are listed and described, including critical system inputs and outputs. A flow- chart 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.

The user interface activities described in Steps 1 and 3 are usually performed con- currently with these steps.

Understanding the User's Mental Model

The next phase in interface design is to thoroughly describe the expected system user or users and their current tasks. The former will be derived from the kinds of informa- tion collected in Step 1 "Understand the User or Client," and the requirements analy- sis techniques described above. 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 rep- resentation of a person's current conceptualization and understanding of something. Mental models are gradually developed in order to understand, explain, and do some- thing. 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 break-ing down the user's activities to the individual task level. The goal is to obtain an un- derstanding of why and how people currently do the things that will be automated. Knowing why establishes the major work goals; knowing how provides details of ac- tions 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 descrip-tion of all user tasks and interactions.

Work activities are studied and/or described by users using the techniques just re- viewed; direct observation, interviews, questionnaires, or obtaining measurements of actual current system usage. Measurements, for example, may be obtained for the fre-quency with which tasks are performed or the number of errors that are made.

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 in-corporated as design iteration occurs. Another result is a list of objects the users see asimportant to what they do. The objects can be sorted into the following categories:

- Concrete objects—things that can be touched.
- People who are the object of sentences—normally organization employees, cus- tomers, 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 ob- jects, and the actions that will be performed. A conceptual model is based on the user'smental model. Since the term mental model refers to a person's current level of knowl- edge about something, people will

always have them. Since mental models are influ- enced by a person's experiences, and people have different experiences, no two user mental models are likely to be exactly the same. Each person looks at the interface from a slightly different perspective.

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 concep- tual 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. As a per-son works with a system, he or she then develops a *mental model of the system*. The sys-tem mental model the user derives is based upon system's behavior, including factors such as the system inputs, actions, outputs (including screens and messages), and its feedback and guidance characteristics, all of which are components of the conceptual model. Documentation and training also play a formative role. Mental models will bedeveloped regardless of the particular design of a system, and then they will be modi-fied with experience. What must be avoided in design is creating for the user a con- ceptual model that leads to the creation of a false mental model of the system, or that inhibits the user from creating a meaningful or efficient mental model.

Guidelines for Designing Conceptual Models

- Reflect the user's mental model, not the designer's.
- Draw physical analogies or present metaphors.
- Comply with expectancies, habits, routines, and stereotypes.
- Provide action-response compatibility.
- Make invisible parts and process of a system visible.
- Provide proper and correct feedback.
- Avoid anything unnecessary or irrelevant.
- Provide design consistency.
- Provide documentation and a help system that will reinforce the conceptual model.

- Promote the development of both novice and expert mental models.

Unfortunately, little research is available to assist the software designer in creating *conceptual models*. Development of a user's mental model can be aided, however, by fol- lowing these general guidelines for conceptual model development.

Reflect the user's mental model, not the designer's. A user will have different ex- pectations 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. The designer's model is focused on the design of the interface, the kinds of objects, the interaction meth- ods, and the visual representations on the screen. Objects must be defined, along with their relationships, behaviors, and properties. Interaction methods must also be defined, such as input mechanisms, interaction techniques, and the contents of menus. Visual screen representations must also be created, including functional-ity and appearance.

Draw physical analogies or present metaphors. Replicate what is familiar and well known. Duplicate actions that are already well learned. The success of graphical systems can be attributed, in part, to their employing the desktop metaphor. A metaphor, to be effective, must be widely applicable within an interface. Metaphors that are only partially or occasionally applicable should not be used. In the event that a metaphor cannot be explicitly employed in a new interface, structure the new interface in terms of familiar aspects from the manual world.

Comply with expectancies, habits, routines, and stereotypes. Create a system that builds on knowledge, habits, routines, and expectancies that already exist. Use fa- miliar associations, avoiding the new and unfamiliar. With color, for example, ac-cepted meanings for red, yellow, and green are already well established. Use words and symbols in their customary ways. Replicate the language of the user, and create icons reflecting already known images.

Provide action-response compatibility. All system responses should be compatiblewith the actions that elicit them. Names of commands, for example, should reflect the actions that will occur. The organization of keys in documentation or help screens should reflect the ordering that actually exists on the keyboard.

Make invisible parts of the system visible. Systems are composed of parts and processes, many of which are invisible to the user. In creating a mental model, a person must make a hypothesis about what is invisible and how it relates to what is visible. New users of a system often make erroneous or incomplete assump- tions about what is invisible and develop a faulty mental model. As more experi-ence is gained, their mental models evolve to become more accurate and complete. Making invisible parts of a system visible will speed up the process of developing correct mental models.

An example of a process being made visible can be illustrated by moving a document between files. In a command language interface, the document must bemoved through a series of typed commands. The file is moved invisibly, and the user assumes correctly, unless an error message is received. In a graphical direct-manipulation system, the entire process is visible, with the user literally picking up the file in one folder by clicking on it, and dragging it to another folder.

Provide Proper and Correct Feedback. Be generous in providing feedback. Keep a person informed of what is happening, and what has happened, at all times, including:

Provide a continuous indication of status. Mental models are difficult to developif things happen, or are completed, unknown to the user. During long pro- cessing sequences, for example, interim status messages such as loading, "opening . . ." or "searching . . ." reassure the user and enable him or her to understand internal processes and more accurately predict how long some- thing will take. Such messages also permit the pinpointing of problems if theyoccur.

Provide visible results of actions. For example, highlight selected objects, display new locations of moved objects, and show files that are closed.

Display actions in progress. For example, show a window that is being changed in size actually changing, not simply the window in its changed form. This will strengthen cause-and-effect relationships in the mental model.

Present as much context information as possible. To promote contextual under- standing, present as much background or historical information as possible. For example, on a menu screen or in navigation, maintain a listing of the choices selected to get to the current point. On a query or search screen, showthe query or search criteria when displaying the results.

Provide clear, constructive, and correct error messages. Incomplete or misleading error messages may cause false assumptions that violate and weaken the user's mental model. Error messages should always be structured to reinforce the mental model. For example, error messages addressing an incomplete ac- tion should specify *exactly* what is missing, not simply advise a person that something is incomplete.

Avoid the unnecessary or irrelevant. Never display irrelevant information on the screen. People may try to interpret it and integrate it into their mental models, thereby creating a false one. Irrelevant information might be unneeded data fields, screen controls, system status codes, or error message numbers. If poten- tially misleading information cannot be avoided, point this out to the user.

Also, do not overuse display techniques, or use them in meaningless ways. Too much color, for example, may distract people and cause them to make erro- neous assumptions as they try to interpret the colors. The result will be a faulty and unclear mental model.

Provide design consistency. Design consistency reduces the number of concepts to be learned. Inconsistency requires the mastery of multiple models. If an occa- sional inconsistency cannot be avoided, explain it to the user. For example, if an error is caused by a user action that is inconsistent with other similar actions, ex-plain in the error message that this condition exists. This will prevent the user from falsely assuming that the model he or she has been operating under is incorrect.

Provide documentation and a help system that will reinforce the conceptual model. Consistencies and metaphors should be explicitly described in the user documentation. This will assist a person in learning the system. 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 sys-tem. It will be easier for novices to form an initial system mental model if they are protected from the full complexity of a system. Employ levels of functionality that can be revealed through progressive disclosure.

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.

All *objects* that have to be manipulated to get work done must be clearly described. Their behavioral characteristics must be established and the attributes that differenti- ate each kind of object must be identified. Relationships of objects to each other and the people using them must be determined. The actions people take on objects must also be described. State information or attributes that each object in the task must preserve, dis- play, or allow to be edited must be defined.

The most important objects must be made very obvious and easy to manipulate. Weinschenk (1995) suggests that if the most important objects are not obvious in the workflow, go through the workflow document highlighting all nouns and verbs asso- ciated with nouns. Frequently appearing nouns are possible major objects. Frequently appearing verbs are actions pointing to possible major objects.

Developing Metaphors

- 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.

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. Select a metaphor or analogy for the defined objects. Choose the analogy that works best for the objects and their actions.

Real-world metaphors are most often the best choice. Replicate what is familiar and well known. Duplicate actions that are already well learned. If a faster or better way ex- ists to do something, however, use it. Use simple metaphors, as they are almost alwaysthe most powerful. Use common metaphors; uniqueness adds complexity. Multiple metaphors may coexist. Use major metaphors even if you can't exactly replicate them visually on the screen. Exactly mimicking the real world does not always aid under- standing. It can lead a person to expect behavioral limitations that do not actually exist. A representation will be satisfactory. Finally, test the selected metaphors. Do they match one's expectations and experiences? Are they easily understood or quickly learned? Change them, if testing deems it necessary.

A common metaphor in a graphical system is the desktop and its components, items such as folders and a trash bin. The Web utilizes a library metaphor for the activities of browsing and searching. Browsing in a library occurs when you wander around book stacks looking for something interesting to read. When searching you devise an activeplan to find some specific information. For example, first, check the topic in the card catalog. Next, ask the librarian, and so forth.

A word of caution in creating metaphors, however. Today's technology permits doing a lot of things, many not even thinkable in the old manual world (or even the old computer world). Do not constrain yourself from developing a more powerful interfacebecause a current metaphor just happens to exist. If you do limit yourself, you may findyourself in the position of the farm tractor

designers of the early last century. In devel-oping a new tractor, the metaphor was the horse and plow. Reins controlled the horse, so reins were installed on the tractor for controlling it as well. Needless to say it was not successful. We do not want to read about you sometime later this century.

The User's New Mental Model

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

What happens, however, if the new system does not accurately reflect the user's ex- isting mental model? The results include breakdowns in understanding, confusion, er-rors, loss of trust, and frustration. Another result is an inability to perform the task or job.Historically, when system designers have known in advance there was a gap be- tween their conceptual model and the mental model the user would bring to the new system, designers have tried to bridge this gap through extensive documentation and training. The problems with this approach are: people are unproductive while being trained, people rarely read the documentation and training materials, and, even if thetraining material is read, the material is presented out of context. This creates difficul-

ties for the users in understanding the material's relevance to their needs and goals.

Design Standards or Style Guides

A design standard or style guide documents an agreed-upon way of doing something. In interface design it describes the appearance and behavior of the interface and pro- vides some

guidance on the proper use of system components. It also defines the in- terface standards, rules, guidelines, and conventions that must be followed in detailed design. It will be based on the characteristics of the system's hardware and software, the principles of good interface and screen design, the needs of system users, and anyunique company or organization requirements that may exist.

Value of Standards and Guidelines

Developing and applying design standards or guidelines achieves 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.

Business System Interface Standards and Guidelines

While some businesses and organizations developed and implemented human- computer interface design standards as far back as the 1970s (for example, see Galitz and DiMatteo, 1974), it was not until the late 1980s the computer industry in general and other end-user organizations, fully awakened to their need. Then, a flurry of guide- line documents began to appear. Some were for internal company or organization use only; others were published for general consumption by companies such as IBM (1987), Sun Microsystems (1990), Apple Computer (1992b), and Microsoft (1992). These guide- lines have been updated over the last decade, and today many of these interface guide-lines are published on the Web as well.

Concurrently government and trade organizations also began working on develop- ing

interface guidelines and standards. Organizations addressing these issues have in-

cluded the International Standards Organization (ISO), the American National Stan- dards Institute (ANSI), and the Human Factors and Ergonomics Society.

Unfortunately, past research on guideline utilization in business systems has hardly been encouraging. Standards conformance problems identified include difficulties in finding information being sought, difficulties in interpreting information, and numer- ous rules violations. Thovtrup and Nielsen (1991), for example, reported that designers were only able to achieve a 71 percent compliance with a two-page standard in a labo- ratory setting. In an evaluation of three real systems, they found that the mandatory rules of the company's screen design standard were violated 32 to 55 percent of the time. Thovtrup and Nielsen, in analyzing why the rules in the screen design standard were broken, found a very positive designer attitude toward the standard, both interms of its value and content. Rules were not adhered to, however, for the following

reasons:

- An alternative design solution was better than that mandated by the standard.
- Available development tools did not allow compliance with the standard.
- Compliance with the standard was planned, but time was not yet available to implement it.
- The rule that was broken was not known or was overlooked.

Web Guidelines and Style Guides

Web interface design issues have also unleashed a plethora of Web-specific design guidelines and style guides, many of which are found on the Web itself. These guide- lines can be seen on the sites of the various computer companies and interface consult-ing firms, in newsletters, and even on personal Web sites. While many of the traditionalinterface guidelines are applicable in a Web environment, the Web does impose a hostof additional considerations.

The haste to publish Web design guidelines has been fueled by the explosive growthof the Web and a corresponding explosive growth in the number of developers creat- ing sites for public access. In the brief existence of the Web, there has not been an op- portunity for conventions and

style guides to be properly developed and then accepted by the development community. This situation is made worse by the fact that many Web developers have had limited knowledge of traditional interface issues and con- cerns, and many are unfamiliar with the traditional interface design guidelines. Web guideline documents have attempted to fill this void.

Since a Web user can freely move among a seemingly endless supply of sites, no site will be seen in isolation. Commonality is of even greater importance than in business systems, where movement occurs between small numbers of applications. Today, many uniquely Web standards and guidelines are evolving by trial and error. Things are being tried to see what works best. De facto standards are being established when an overwhelming majority of big sites focus on one way to do something. An exampleis a menu bar that now frequently appears down the left side of the page. Standards and conventions will continue to evolve with experience and as the results of usability research become available. Worldwide standards are also being looked at by organi- zations such as the World Wide Web Consortium (2001).

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 alterna- tives are appropriate.
- Include concrete examples of correct design.
- Design the guideline document following recognized principles for good documentdesign.
- Provide good access mechanisms such as a thorough index, a table of contents, glos- saries, and checklists.

Checklists and rationale. Provide checklists for presenting key principles and guide- lines. Checklists permit ease in scanning, ease in referring to key points, and makea document more readable by breaking up long sequences of text. Also provide a rationale for why the particular guidelines should be used. Understanding the reasoning will increase guideline acceptance. This is

especially important if the guideline is a deviation from a previous design practice. Also, when two or more design alternatives exist, provide a rationale describing the conditions under which the alternatives are appropriate. It may not be easy for designers to infer when various alternatives are appropriate. You have probably noticed that this book uses a checklist format to present key guidelines and thoughts, and guideline rationale is described in the text.

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

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

Design Support and Implementation

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

- Begin applying the guidelines immediately.

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

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

Applying the Guidelines. Two questions often asked are, "Is it too late to develop and implement

standards?" and "What will be the impact on systems and screensnow being used?" To address these questions, researchers reformatted several al-phanumeric inquiry screens to improve their comprehensibility and readability. When these reformatted screens were presented to expert system users, decision- making time remained the same but errors were reduced. For novice system users, the reformatted screens brought large improvements in learning speed and accuracy. Therefore, it appears, that changes that enhance screens will benefit *both* novice and expert users already familiar with the current screens. It is nevertoo late to begin to change.