MODULE 1

Chapter 1: User Interactive Systems

INTRODUCTION

User interface (UI) design is **the process designers use to build interfaces in software or computerized devices, focusing on looks or style**. Designers aim to create interfaces which users find easy to use and pleasurable. UI design refers to graphical user interfaces and other forms—e.g., voice-controlled interfaces.

Usability Requirements

Every designer wants to build high-quality interfaces that are admired by colleagues, celebrated by users, and imitated frequently. Appreciation comes not from flamboyant promises or stylish advertising, but rather from inherent quality features such as usability, universality and usefulness.

These goals are achieved through thoughtful planning, sensitivity to user needs, devotion to requirements analysis, and diligent testing.

Managers can promote attention to user-interface design by selection of personnel, preparation of schedules and milestones, construction and application of guidelines documents, and commitment to testing. Designers then propose multiple design alternatives for consideration, and the leading contenders are subjected to further development and testing

Successful designers go beyond the vague notion of "user friendliness," probing deeper than simply making a checklist of subjective guidelines

Effective interfaces generate positive feelings of success, competence, mastery, and clarity in the user community.

Setting explicit goals helps designers to achieve those goals. In getting beyond the vague quest for user-friendly systems, managers and designers can focus on specific goals that include well-defined system engineering and measurable human-factors objectives. The U.s. Military Standard for Human Engineering Design Criteria (1999) states these purposes:

• Achieve required performance by operator, control, and maintenance personnel

• Minimize skill and personnel requirements and training time

• Achieve required reliability of personnel-equipment/software combinations

• Foster design standardization within and among systems

These functional purposes are good starting points, but effective interfaces might also enhance the quality of life for users or improve their communities.

The first goal in requirements analysis is to ascertain the users/ needs-that is, what tasks and subtasks must be carried out. The frequent tasks are easy to determine, but the occasional tasks, the exceptional tasks for emergency conditions, and the repair tasks to cope with errors in use of the interface are more difficult to discover. A vital second goal is to ensure proper reliability: actions must function as specified, displayed data must reflect the database contents, and updates must
be applied correctly.

A vital second goal is to ensure proper reliability: actions must function as specified, displayed data must reflect the database contents, and updates must be applied correctly.

The third set of goals for designers is to consider the context of use and promote appropriate standardization, integration, consistency, and portability. As the number of users and software packages increases, the pressures for and benefits of standardization grow.

Standardization refers to common user-interface features across multiple applications. Apple Computers 0992,2002) successfully developed an early standard that was widely applied by thousands of developers, enabling users to learn multiple applications quickly.

Integration across application packages and software tools was one of the key design principles of Unix. (Portability across hardware platforms was another.) If file formats are used consistently, users can apply multiple applications to transform, refine, or validate their data.

Consistency primarily refers to common action sequences, terms, units, layouts, colors, typography, and so on within an application program. Consistency is a strong determinant of success of interfaces. It is naturally extended to include compatibility across application programs and compatibility with paper or non-computer-based systems. Compatibility across versions is a troubling demand, since the desire to accommodate novel functionality or improved designs competes with the benefits of consistency.

Portability refers to the potential to convert data and to share user interfaces across multiple software and hardware environments. Arranging for portability is a challenge for designers, who must contend with different display sizes and resolutions, color capabilities, pointing devices, data formats, and so on.

The fourth goal for interface designers is to complete projects on schedule and within budget. Delayed delivery or cost overruns can threaten an interface project because of the confrontational political atmosphere in a company, or because the competitive market environment contains potentially overwhelm• ing forces. If an in-house system is delivered late, other projects may be affected, and the disruption may cause managers to choose to install an alternative system.

Usability Measures

If adequate requirements are chosen, reliability is ensured, standardization is addressed, and scheduling and budgetary planning are complete, developers can focus their attention on the design and testing process.
Eventually a touchscreen interface with reduced functionality and better information presentation was developed and became a big success in the public reading rooms.
The next step in evolution was the development of a World Wide Web version of the catalog to allow users anywhere in the world to access the catalog and other databases
Careful determination of the user community and of the benchmark set of tasks is the basis for establishing usability goals and measures.
The ISO 9241 standard focuses on admirable goals *(effectiveness, efficiency,* and *satisfaction),* but the following usability measures, which focus on the latter two goals, lead more directly to practical evaluation:

- ***Time to learn***. How long does it take for typical members of the user community to learn how to use the actions relevant to a set of tasks?
- ***Speed of performance***. How long does it take to carry out the benchmark tasks?
- ***Retention over time***. How well do users maintain their knowledge after an hour, a day, or a week? Retention may be linked closely to time to learn, and frequency of use plays an important role.
- ***Subjective satisfaction***. How much did users like using various aspects of the interface? The answer can be ascertained by interview or by written surveys that include satisfaction scales and space for free-form comments.
- ***Rate of errors by users***: How many and what kinds of errors do people make in carrying out the benchmark tasks? Although time to make and correct errors might be incorporated into the speed of performance, error handling is such a critical component of interface usage that it deserves extensive study.

Every designer would like to succeed in every category, but there are often forced tradeoffs.

If the rate of errors is to be kept extremely low, speed of performance may have to be sacrificed. In some applications, subjective satisfaction may be the key determinant of success; in others, short learning times or rapid performance may be paramount.

After multiple design alternatives have been raised, the leading possibilities should be reviewed by designers and users.

Usability Motivations

The enormous interest in interface usability arises from the growing recognition of how poorly designed many current interfaces are and of the benefits elegant interfaces bring to users.

This increased motivation emanates from developers of life-critical systems; industrial and commercial systems; office, home, and entertainment applications; exploratory, creative, and collaborative interfaces; and sociotechnical systems.

Life-critical systems

- Life-critical systems include those that control air traffic, nuclear reactors, power utilities, police or fire dispatch, military operations, and medical instruments
- In these applications high costs are expected, but they should yield high reliability and effectiveness.
- Subjective satisfaction is less of an issue because the users are well-motivated professionals. Retention is obtained by frequent use of common functions and practice sessions for emergency actions.

Industrial and commercial uses

- Typical industrial and commercial uses include banking, insurance, order entry, inventory management, airline and hotel reservations car rentals, utility billing, credit-card management, and point-of-sales terminals.
- Operator training time is expensive, so ease of learning is important.
- Subjective satisfaction is of modest importance; retention is obtained by frequent use. Speed of performance becomes central for most of these applications because of the high volume of transactions, but operator fatigue, stress, and burnout are legitimate concerns.

Office, home, and entertainment applications

- The rapid expansion of office, home, and entertainment applications is the third source of interest in usability.
- For these interfaces, ease of learning, low error rates, and subjective satisfaction are paramount because use is frequently discretionary and competition is fierce.
- Novices are best served by a constrained, simple set of actions, but as users' experience increases, so does their desire for more extensive functionality and rapid performance.

Exploratory, creative, and collaborative interfaces

- An increasing fraction of computer use is dedicated to supporting human intellectual and creative enterprises.
- Exploratory applications include World Wide Web browsing, search engines, scientific simulation, and business decision making.
- Collaborative interfaces enable two or more people to work together, even if the users are separated by time and space, through use of electronic text, voice, and video mail; through electronic meeting systems that facilitate face-to-face meetings; or through groupware that enables remote collaborators to work concurrently on a document, map, spreadsheet, or image.

Sociotechnical systems

- A growing domain for usability is in complex systems that involve many people over long time periods, such as systems for voting, health support, identity verification, and crime reporting.
- Interfaces for these systems, often created by governmental organizations, have to deal with trust, privacy, and responsibility, as well as limiting the harmful effects of malicious tampering, deception, and incorrect information.
- Users want access to verifiable sources, adequate feedback about their actions, and ways of easily checking status.
- Designers of sociotechnical systems have to take into consideration the diverse levels of expertise that users with different roles have.
- Designs for the professional administrators and the seasoned investigators will enable rapid performance of complex procedures with visualization tools to spot unusual patterns or detect fraud in usage logs.

Universal Usability

The remarkable diversity of human abilities, backgrounds, motivations, personalities, cultures, and work styles challenges interface designers.

Understanding the physical, intellectual, and personality differences between users is vital for expanding market share, supporting required government services, and enabling creative participation by the broadest possible set of users.

Variations in physical abilities and physical workplaces

- Accommodating diverse human perceptual, cognitive, and motor abilities is a challenge to every designer.
- Basic data about human dimensions comes from research in *anthropometry*
- Thousands of measures of hundreds of features of people-male and female, young and adult, European and Asian, underweight and overweight, and tall and short-provide data to construct means and 5- to 95-percentile groupings.

- Head, mouth, neck, shoulder, chest, arm, hand, finger, leg, and foot sizes have been carefully cataloged for a variety of populations.
- The great diversity in these static measures reminds us that there can be no image of an "average" user, and that compromises must be made or multiple versions of a system must be constructed.
- These physical abilities influence elements of the interactive-system design. They also playa prominent role in the design of the workplace or workstation (or playstation). The draft standard *Human Factors Engineering of Computer Workstations* (2002) lists these concerns:
  - Work Surface and display-support height
  - Clearance under work surface for legs
  - Work-surface width and depth
  - Adjustability of heights and angles for chairs and work surfaces
  - Posture seating depth and angle; back-rest height and lumbar support
  - Avalibilty of armrests, footrests, and palmrests
- Workplace design is important in ensuring high job satisfaction, high performance, and low error rates.
- The most elegant screen design can be compromised by a noisy environment, poor lighting, or a stuffy room, and that compromise will eventually lower performance, raise error rates, and discourage even motivated users.
- The physical design of workplaces is often discussed under the term *ergonomics.* Anthropometry, sociology, industrial psychology, organizational behavior studies, and anthropology may offer useful insights in this area.

Diverse cognitive and perceptual abilities

- A vital foundation for interactive-systems designers is an understanding of the cognitive and perceptual abilities of the users.
- The human ability to interpret sensory input rapidly and to initiate complex actions makes modern computer systems possible.
- The journal *Ergonomics Abstracts* offers this classification of human cognitive processes:

- Short-term and working memory
- Long-term and semantic memory
- Problem solving and reasoning
- Decision making and risk assessment
- Language communication and comprehension
- Search, imagery, and sensory memory
- Learning, skill development, knowledge acquisition, and conceptattainment

- They also suggest this set of factors affecting perceptual and motor performance:
    - Arousal and vigilance
    - Fatigue and sleep deprivation
    - Perceptual (mental) load
    - Knowledge of results and feedback
    - Monotony and boredom
    - Sensory deprivation
    - Nutrition and diet
    - Fear, anxiety, mood, and emotion
    - Drugs, smoking, and alcohol
    - Physiological rhythms
- These vital issues are not discussed in depth in this book, but they have a profound influence on the quality of the design of most interactive systems

Personality differences

- Some people dislike computers or are made anxious by them; others are attracted to or are eager to use computers.
- Even people who enjoy using computers may have very different preferences for interaction styles, pace of interaction, graphics versus tabular presentations, dense versus sparse data presentation, step-by-step work versus all-at-once work, and so on.
- Designers can get into lively debates speculating why women prefer these games, which are distinguished by their less violent action and quieter soundtracks.
- Turning from games to productivity tools, the largely male designers may not realize the effects on women users when command names require the users to KILL a process or ABORT a program.
- Unfortunately, there is no simple taxonomy of user personality types. A pop• ular, but controversial, technique is to use the Myers-Briggs Type Indicator (MBTI) (Keirsey, 1998), which is based on Carl lung's theories of personality types. lung conjectured that there were four dichotomies:
  - ***Extroversion versus introversion.*** Extroverts focus on external stimuli and like variety and action, whereas introverts prefer familiar patterns, rely on their inner ideas, and work alone contentedly.
  - ***Sensing versus intuition.*** Sensing types are attracted to established routines, are good at precise work, and enjoy applying known skills, whereas intuitive types like solving new problems and discovering new relations but dislike taking time for precision.
  - ***Perceptive versus judging***. Perceptive types like to learn about new situations but may have trouble making decisions, whereas judging types like to make a careful plan and will seek to carry through the plan even if new facts change the goal.

- ***Feeling versus thinking***. Feeling types are aware of other people's feelings, seek to please others, and relate well to most people, whereas thinking types are unemotional, may treat people impersonally, and like to put things in logical order.

Cultural and international diversity

Another perspective on individual differences has to do with cultural, ethnic, racial, or linguistic background Users who were raised learning to read Japanese or Chinese will scan a screen differently from users who were raised learning to read English or French.

More and more is being learned about computer users from different cultures, but designers are still struggling to establish guidelines for designing for multiple languages and cultures. User-interface design concerns for internationalization include the following:

- Characters, numerals, special characters, and diacriticals
- Left-to-right versus right-to-Ieft versus vertical input and reading
- Date and time formats
- Numeric and currency formats
- Weights and measures
- Telephone numbers and addresses
- Names and titles (Mr., Ms., Mme., M., Dr.)
- Social security, national identification, and passport numbers
- Capitalization and punctuation
- Sorting sequences
- Icons, buttons, and colors
- Pluralization, grammar, and spelling
- Etiquette, policies, tone, formality, and metaphors

The list is long and yet incomplete. Whereas early designers were often excused from cultural and linguistic slips, the current highly competitive atmosphere means that more effective localization may produce a strong advantage. To promote effective designs, companies should run usability studies with users from different countries, cultures, and language communities.

The role of information technology in international development is steadily growing, but much needs to be done to accommodate the diverse needs of users with vastly different language skills and technology access.

Users with disabilities

The flexibility of desktop and web software makes it possible for designers to provide special services to users who have disabilities

Screen magnification to enlarge portions of a display or text conversion to Braille or voice output can be done with hardware and software supplied by many vendors

Text-to-speech conversion can help blind users to receive electronic mail or to read text files, and speech-recognition devices permit voice-controlled operation of some software.

Graphical user interfaces were a setback for vision-impaired users, but technology innovations from commercial tools such as Freedom Scientific's JAWS, GW Micro's Window-Eyes, or Dolphin's HAL screen readers facilitate conversion of spatial information into spoken text.

Users with hearing impairments generally can use computers with only simple changes (conversion of tones to visual signals is often easy to accomplish) and can benefit from office environments that make heavy use of electronic mail and facsimile transmission (FAX).

Designers can benefit by planning early to accommodate users who have disabilities, since at this point substantial improvements can be made at low or no cost.
Further information about accommodation in workplaces, schools, and the home is available from many sources:

- Private foundations (e.g., the American Foundation for the Blind and the National Federation of the Blind)
- Associations (e.g., the Alexander Graham Bell Association for the Deaf, the National Association for the Deaf, and the Blinded Veterans Association)
- Government agencies (e.g., the National Library Service for the Blind and Physically Handicapped of the Library of Congress and the Center for Technology in Human Disabilities at the Maryland Rehabilitation Center)
- University groups (e.g., the Trace Research and Development Center on Communications and the Control and Computer Access for Handicapped Individuals at the University of Wisconsin)
- Manufacturers (e.g., Apple, IBM, Microsoft, and Sun Microsystems)

Based on observations of 62 students using 26 packages over 5.5 months, Neuman's advice to designers of courseware for learning-disabled students is applicable to all users:

- Present procedures, directions, and verbal content at levels and in formats that make them accessible even to poor readers.
- Ensure that response requirements do not allow students to complete programs without engaging with target concepts.
- Design feedback sequences that explain the reasons for students' errors and that lead students through the processes necessary for responding correctly.
- Incorporate reinforcement techniques that capitalize on students' sophistication with out-of-school electronic materials.

Older adult users

- Understanding the human factors of aging can lead us to computer designs that will facilitate access by older adult users.

- The benefits to senior citizens include improved chances for productive employment and opportunities to use writing, accounting, and other computer tools, plus the satisfactions of education, entertainment, social interaction, and challenge.
- The benefits to society include increased access to seniors for their experience and the emotional port they can provide to others.
- The further good news is that interface designers can do much to accommodate older adult users, and thus to give older adults access to the beneficial aspects of computing and network communication.
- As the world's population grows older, designers in many fields are adapting their work to serve older adult citizens. Larger street signs, brighter traffic lights, and better nighttime lighting can make driving safer for drivers and pedestrians.

Designing for children

- Another lively community of users is children, whose uses emphasize entertainment and education. Even pre-readers can use computer-controlled toys, music generators, and art tools.
- As they mature to begin reading and gain limited keyboard skills, they can use a wider array of portable devices, desktop applications, and web services.
- The noble aspirations of developers of children's software include educational acceleration, socialization with peers, and the positive self-image or self confidence that comes from mastery of skills.
- Designers also have to find the balance between children's desire for challenge and parents' requirements for safety.
- Designers of children's software also have a responsibility to attend to dangers, especially in web-based environments, where parental control over access to violent, racist, or pornographic materials is unfortunately necessary.
- The capacity for playful creativity in art, music, and writing, as well as educational activities in science and math, remain potent reasons to pursue children's software.
- Offering access to educational materials from libraries, museums, government agencies, schools, and commercial sources enriches their learning experiences and serves as a basis for children to construct their own web resources, participate in collaborative efforts, and contribute to community-service projects.
- Providing programming and simulation-building tools enables older children to take on complex cognitive challenges and construct ambitious artifacts for others to use.

Accommodating hardware and software diversity

- In addition to accommodating different classes of users and skill levels, designers need to support a wide range of hardware and software platforms.

- The rapid progress of technology means that newer systems may have a hundred or a thousand times greater storage capacity, faster processors, and higher-bandwidth networks.

- The challenge of accommodating diverse hardware is coupled with the need to ensure access through many generations of software.

For at least the next decade, three of the main technical challenges will be:

- ***Producing satisfying and effective Internet interaction on high-speed (broadband) and slower (dial-up and some wireless) connections***. Some technology breakthroughs have already been made in compression algorithms to reduce file sizes for images, music, animations, and even video, but more are needed. New technologies are needed to enable pre-fetching or scheduled downloads. User control of the amount of material downloaded for each request could also prove beneficial (for example, allowing users to specify that a large image should be reduced to a smaller size, sent with fewer colors, converted to a simplified line drawing, or even replaced with just a text description).
- ***Enabling access to web services from large displays (1200 x 1600 pixels or larger) and smaller mobile devices (640 x 480 and smaller).*** Rewriting each web page for different display sizes may produce the best quality, but this approach is probably too costly and time-consuming for most web providers. New software-tool breakthroughs are needed to allow web-site developers to specify their content in such a way that automatic conversions can be made for an increasing range of display sizes.

- ***Supporting easy maintenance of or automatic conversion to multiple languages.*** Commercial operators recognize that they can expand their markets if they can provide access in multiple languages and across multiple countries. This means isolating text to allow easy substitution, choosing appropriate metaphors and colors, and addressing the needs of diverse cultures

Goals for Our Profession

- Clear goals are useful not only for interface development but also for educational and professional enterprises.
- Three broad goals seem attainable:

1. influencing academic and industrial researchers
2. providing tools, techniques, and knowledge for commercial developers
3. raising the computer consciousness of the general public

Influencing academic and industrial researchers

- Early research in human-computer interaction was done largely by introspection and intuition, but this approach suffered from lack of validity, generality, and precision.

- The scientific method for interface research, which is based on controlled experimentation, has this basic outline:
    - Understanding of a practical problem and related theory
    - Lucid statement of a testable hypothesis
    - Manipulation of a small number of independent variables
    - Measurement of specific dependent variables
    - Careful selection and assignment of subjects
    - Control for bias in subjects, procedures, and materials
    - Application of statistical tests
    - Interpretation of results, refinement of theory, and guidance for experimenters
- There are so many fruitful directions for research that any list can be only a provocative starting point. Here are a few:

- ***Reduced anxiety and fear of computer usage****:* Although computers are widely used, they still serve only a fraction of the population. Many otherwise competent people resist use of computers. Some older adults avoid helpful computer-based devices, such as bank terminals or word processors, because they are anxious about-or even fearful of-breaking the computer or making an embarrassing mistake.
- ***Graceful evolution****:* Although novices may begin their interactions with a computer by using menu selection, they may wish to evolve to faster or more powerful facilities. Methods are needed to smooth the transition from novice to knowledgeable user to expert. The differing requirements of novices and experts in prompting, error messages, online assistance, display complexity, locus of control, pacing, and informative feedback all need investigation.
- ***Specification and implementation of interaction****:* User-interface building tools reduce implementation times by an order of magnitude when they match the task. Advanced research on tools to aid interactive-systems designers and implementers might have substantial payoffs in reducing costs and improving quality.
- ***Direct manipulatiOlz****:* Visual interfaces in which users operate on a representation of the objects of interest are extremely attractive Newer forms of direct manipulation-such as visual languages, information visualization, telepresence, and virtual reality-are further topics for research.
- ***Input devices****:* The plethora of input devices presents opportunities and challenges to interface designers Such conflicts could be resolved through experimentation with multiple tasks and users. Underlying issues include speed, accuracy, fatigue, error correction, and subjective satisfaction.
- ***Online help****:* Although many interfaces offer some help or tutorial information online, we have only limited understanding of what constitutes effective

design for novices, knowledgeable users, and experts.

- ***Information exploration**:* As navigation, browsing, and searching of multimedia digital libraries and the World Wide Web become more common, the pressure for more effective strategies and tools will increase. Large databases of text, images, graphics, sound, and scientific data will become easier to explore with emerging information-visualization tools.

Providing tools, techniques, and knowledge for commercial developers

- User-interface design and development are hot topics, and international competition is lively.
- Employers, who used to see usability as a secondary topic, are increasingly hiring user-interface designers, information architects, user-interface implementers, and usability testers.
- These employers recognize the competitive advantage from high-quality consumer interfaces and from improving the performance of their employees.
- Guidelines documents have been written for general and specific audiences.
- Iterative usability studies and acceptance testing are appropriate during interface development. Once the initial interface is available, refinements can be made on the basis of online or printed surveys, individual or group interviews, or more controlled empirical tests of novel strategies
- Feedback from users during the development process and for evolutionary refinement can provide useful insights and guidance. Online electronic-mail facilities allow users to send comments directly to the designers. Online user consultants and fellow users can provide prompt assistance and supportive encouragement.

Raising the computer consciousness of the generalpublic

- The media are so filled with stories about computers that raising public consciousness of these tools may seem unnecessary. However, many people are still uncomfortable with computers
- One of our goals is to encourage users to translate their internal fears into outraged action.
- Instead of feeling guilty when they get a message such as SYNTAX ERROR, users should express their anger at the interface designer who was so inconsiderate and thoughtless.
- Usability ultimately becomes a question of national priorities. Advocates of electronic voting and other services, promoters of e-healthcare, and visionaries of e-Iearning increasingly recognize the need to influence allocation of government resources and commercial research agendas.

Chapter 2: Guidelines, Principles and Theories

INTRODUCTION

Successful designers of interactive systems know that they can and must go beyond intuitive judgments made hastily when a design problem emerges. Fortunately, guidance for designers is available in the form of

1. specific and practical guidelines
2. middle-level principles
3. high-level theories and models

The practical guidelines prescribe cures for design problems, caution against dangers, and provide helpful reminders based on accumulated wisdom. The middle-level principles help in analyzing and comparing design alternatives

Guidelines

Interface designers have tried to write down guidelines torecord their insights and guide the efforts of future designers.

A guidelines document helpsby developing a shared language and then promoting consistency among multiple designers in terminology, appearance, and action sequences.

The examples address some key topics, but they merely sample the thousands of guidelines that have been written.

Navigating the interface

Since navigation can be difficult for many users, providing clear rules is helpful.

This sample of guidelines comes from the National Cancer Institute's effort to assist government agencies with design of informative web pages, but these guidelines have widespread application
This sample of the guidelines gives useful advice and a taste of their style:

- ***Standardize task sequences***. Allow users to perform tasks in the same sequenceand manner across similar conditions.
- ***Ensure that embedded links are descriptive***. When using embedded links, the linktext should accurately describe the link's destination.
- ***Use unique and descriptive headings***. Use headings that are unique from one

another and conceptually related to the content they describe.

- ***Use check boxes for binary choices***. Provide a check box control for users to make a choice between two clearly distinguishable states, such as "on" or "off."
- ***Develop pages that will print properly***. If users are likely to print one or more pages, develop pages with widths that print properly.
- ***Use thumbnail images to preview larger images***. When viewing full-size images is not critical, first provide a thumbnail of the image.

The goal of these guidelines is to have web-page designers use features that permit users with disabilities to employ screen readers or other special technologies to give them access to web-page content.

These guidelines are clarified by examples and supported by research studies. A goal for guidelines writers is to be clear and comprehensible, using meaningful examples. However, controversies over guidelines are lively, often leading to revisions and the creation of alternatives.

A few of the Priority 1 Accessibility Guidelines are:

- Provide a text equivalent for every nontext element including images, graphical represen• tations of text (including symbols), image map regions, animations (such as animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video.
- For any time-based multimedia presentation (for example, a movie or animation), synchronize equivalent alternatives, such as captions or auditory descriptions of the visual track, with the presentation.
- Ensure that all information conveyed with color is also available without color- for example, from context or markup.
- Title each frame to facilitate frame identification and navigation.

Organizing the display

Display design is a large topic with many special cases. Smith and Mosier (1986)offer five high-level goals as part of their guidelines for data display:

1. ***Consistency of data display***. During the design process, the terminology, abbreviations, formats, colors, capitalization, and so on should all be standardized and controlled by use of a written (or computer-managed) dictionary of these items.
2. ***Efficient information assimilation by the user***. The format should be familiar to the operator and should be related to the tasks required to be performed with the data. This objective is served by rules for neat columns

of data, left justification for alphanumeric data, right justification of integers, lining up of decimal points, proper spacing, use of comprehensible labels, and appropriate measurement units and numbers of decimal digits.

3. ***Minimal memory load on the user***. Users should not be required to remember information from one screen for use on another screen. Tasks should be arranged such that completion occurs with few actions, minimizing the chance of forgetting to perform a step. Labels and common formats should be provided for novice or intermittent users.
4. ***Compatibility of data display with data entry***. The format of displayed information should be linked clearly to the format of the data entry. Where possible and appropriate, the output fields should also act as editable input fields.
5. ***Flexibility for user control of data display***. Users should be able to get the information from the display in the form most convenient for the task on which they are working. For example, the order of columns and sorting of rows should be easily changeable by the users.

These generic guidelines, which emerged from a report on design of control rooms for electric-power utilities (Lockheed, 1981), remain valid:

- Be consistent in labeling and graphic conventions.
- Standardize abbreviations.
- Use consistent formatting in all displays (headers, footers, paging, menus, and so on).

- Be consistent in labeling and graphic conventions.
- Standardize abbreviations.
- Use consistent formatting in all displays (headers, footers, paging, menus, and so on).

Getting the user's attention

Since substantial information may be presented to users for the normal performance of their work, exceptional conditions or time-dependent information must be presented so as to attract attention.

These guidelines detail several techniques for getting the user's attention:

- *Intensity*. Use *two* levels only, with limited use of high intensity to draw attention.
- *Marking*. Underline the item, enclose it in a box, point to it with an arrow, or use an indicator such as an asterisk, bullet, dash, plus sign, or X.
- *Size*. Use up to four sizes, with larger sizes attracting more attention.
- *Choice offonts*. Use up to three fonts.
- *Inverse video*. Use inverse coloring.
- *Blinking*. Use blinking displays or blinking color changes with great care and in

limited areas.

- *Color*. Use up to four standard colors, with additional colors reserved for occasional use.
- *Audio*. Use soft tones for regular positive feedback and harsh sounds for rare emergency conditions.

Some web designers use blinking advertisements or animated icons to attract attention, but users almost universally disapprove. Animation is appreciated primarily when it provides meaningful information, such as for a progress indicator.

Audio tones, like the clicks in keyboards or ringing sounds in telephones, can provide informative feedback about progress. Alarms for emergency conditions do alert users rapidly, but a mechanism to suppress alarms must be provided.

Facilitating data entry

Data-entry tasks can occupy a substantial fraction of the users' time and can be the source of frustrating and potentially dangerous errors. Smith and Mosier (1986) offer five high-level objectives as part of their guidelines for data entry:

1. ***Consistency of data-entry transactions***. Similar sequences of actions should be used under all conditions; similar delimiters, abbreviations, and so on should be used.
2. ***Minimal input actions by user***. Fewer input actions mean greater operator productivity and-usually-fewer chances for error. Making a choice by a single keystroke, mouse selection, or finger press, rather than by typing in a lengthy string of characters, is potentially advantageous. Selecting from a list of choices eliminates the need for memorization, structures the decision-making task, and eliminates the possibility of typographic errors. However, if users must move their hands from a keyboard to a separate input device, the advantage is negated, because home-row position is lost. Expert users often prefer to type six to eight characters instead of moving to a mouse, joystick, or other selection device.

   A second aspect of this guideline is that redundant data entry should be avoided. It is annoying for users to enter the same information in two locations, since the double entry is perceived as a waste of effort and an opportunity for error. When the same information is required in two places, the system should copy the information for the user, who should still have the option of overriding by retyping.
3. ***Minimal memory load all lIsers***. When doing data entry, users should not be required to remember lengthy lists of codes and complex syntactic command strings.
4. ***Compatibility of data entry with data display***. The format of data-entry information should be linked closely to the format of displayed information.
5. ***Flexibility for user control of data entry***. Experienced data-entry operators may prefer to enter information in a sequence that they can control. For example, on some

occasions in an air-traffic-control environment, the arrival time is the prime field in the controller's mind; on other occasions, the altitude is the prime field. However, flexibility should be used cautiously, since it goes against the consistency principle.

PRINCIPLES

While guidelines are narrowly focused, principles tend to be more fundamental, widely applicable, and enduring. However, they also tend to need more clarification.

For example, the principle of recognizing user diversity makes sense to every designer, but it must be thoughtfully interpreted.

Determine users' skill levels

- It is a simple idea but a difficult and, unfortunately, often undervalued goal.
- No one would argue against this principle, but many designers assume that they understand the users and users' tasks. Successful designers are aware that other people learn, think, and solve problems in different ways.
- All design should begin with an understanding of the intended users, including population profiles that reflect age, gender, physical and cognitive abilities, education, cultural or ethnic background, training, motivation, goals, and personality.
- In addition to these profiles, an understanding of users' skills with interfaces and with the application domain is important. Users might be tested for their familiarity with interface features such as traversing hierarchical menus or drawing tools.

For example, a generic separation into novice or first-time, knowledgeable intermittent, and expert frequent users might lead to these differing design goals:

Novice or first-time users: By contrast, first-time users are professionals who know the task concepts but have shallow knowledge of the interface concepts.

- Both groups of users may arrive with learning-inhibiting anxiety about using computers. Overcoming these limitations, via instructions, dialog boxes, and online help, is a serious challenge to the designer of the interface. Restricting vocabulary to a small number of familiar, consistently used concept terms is essential to begin developing the user's knowledge.
- The number of actions should also be small, so that novice and first-time users can carry out simple tasks successfully and thus reduce anxiety, build confidence, and gain positive reinforcement.
- Informative feedback about the accomplishment of each task is helpful, and constructive, specific error messages should be provided when users make mistakes.

Knowledgeable intermittent users: Many people are knowledgeable but intermittent users of a variety of systems-for example, corporate managers using word processors to create templates for travel reimbursements.

- They have stable task concepts and broad knowledge of interface concepts, but they may have difficulty retaining the structure of menus or the location of features.
- Consistent sequences of actions, mean• ingful messages, and guides to frequent patterns of usage will help knowledgeable intermittent users to rediscover how to perform their tasks properly.
- Protection from danger is necessary to support relaxed exploration of features or usage of partially forgotten action sequences.
- These users will benefit from context dependent help to fill in missing pieces of task or interface knowledge. Well organized reference manuals are also useful.

Expert frequent lIsers: Expert "power" users are thoroughly familiar with the task and interface concepts and seek to get their work done quickly. They demand rapid response times, brief and nondistracting feedback, and the shortcuts to carry out actions with just a few keystrokes or selections.

When a sequence of three or four actions is performed regularly, frequent users are willing to create a *macro* or other abbreviated form to reduce the number of steps. Strings of commands, shortcuts through menus, abbreviations, and other accelerators are requirements.

Identify the tasks

- After carefully drawing the user profile, the developers must identify the tasks to be carried out. Every designer would agree that the set of tasks must be determined before design can proceed, but too often the task analysis is done informally or implicitly.
- This helps designers to understand task frequencies and sequences and make the tough decisions about what tasks to support.
- High-level task actions can be decomposed into multiple middle-level task actions, which can be further refined into atomic actions that users execute witha single command, menu selection, and so on.
- Choosing the most appropriate set of atomic actions is a difficult task. If the atomic actions are too small, the users will become frustrated by the large number of actions necessary to accomplish a higher-level task. If the atomic actions are too large and elaborate, the users will need many such actions with special options, or they will not be able to get exactly what they want from the system.

Relative frequency of use is one of the bases for making architectural design decisions. For example, in a word processor:

- Frequent actions might be performed by special keys, such as the four cursor arrows, Insert, and Delete.
- Less frequent actions might be performed by a single letter plus the Orl key, or by a selection from a pull-down menu-examples include underscore, bold, or save.
- Infrequent actions or complex actions might require going through a sequence of menu selections or form filling-for example, to change the printing format or to revise network-protocol parameters.

Choose an interaction style

When the task analysis is complete and the task objects and actions have been identified, the designer can choose from these primary interaction styles: direct manipulation, menu selection, form filling, command language, and natural language

Direct manipulation:

- When a clever designer can create a visual representation of the world of action, the users' tasks can be greatly simplified, because direct manipulation of familiar objects is possible.
- Keyboard entry of commands or menu choices is replaced by use of pointing devices to select from a visible set of objects and actions.
- Direct manipulation is appealing to novices, is easy to remember for intermittent users, and, with careful design, can be rapid for frequent users.

Menu selection:

- In menu-selection systems, users read a list of items, select the one most appropriate to their task, and observe the effect.
- If the terminology and meaning of the items are understandable and distinct, users can accomplish their tasks with little learning or memorization and just a few actions.
- The greatest benefit may be that there is a clear structure to decision making, since all possible choices are presented at one time.
- This interaction style is appropriate for novice and intermittent users and can be appealing to frequent users if the display and selection mechanisms are rapid.

Form Filling:

- When data entry is required, menu selection alone usually becomes cumbersome, and form filling is appropriate.
- Users see a display of related fields, move a cursor among the fields, and enter data where desired. With the form filling interaction style, users must understand the field labels, know the permissible values and the data-entry method, and be capable of responding to error messages.
- This interaction style is most appropriate for knowledgeable intermittent users or frequent users.

Command language:

- For frequent users, command languages provide a strong feeling of being in control. Users learn the syntax and can often express complex possibilities rapidly, without having to read distracting prompts.
- Error rates are typically high, training is necessary, and retention may be poor.
- Error messages and online assistance are hard to provide because of the diversity of possibilities and the complexity of mapping from tasks to interface concepts and syntax.
- Command languages and lengthier query or programming languages are the domain of expert frequent users, who often derive great satisfaction from mastering a complex set of semantics and syntax.

Natural language:

The hope that computers will respond properly to arbitrary natural-language sentences or phrases engages many researchers and system developers, in spite of limited success thus far.

Natural-language interaction usually provides little context for issuing the next command, frequently requires clarification dialog, and may be slower and more cumbersome than the alternatives.

Use the eight golden rules of interface design

This section focuses attention on eight principles, called "golden rules," that are applicable in most interactive systems.

No list such as this can be complete, but it has been well received as a useful guide to students and designers

Strive for consistency: This rule is the most frequently violated one, but following it can be tricky because there are many forms of consistency. Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent color, layout, capitalization, fonts, and so on should be employed through out.

Cater to Universal Usability: Recognize the needs of diverse users and design for *plasticity*, facilitating transformation of content. Novice-expert differences, age ranges, disabilities, and technology diversity each enrich the spectrum ofrequirements that guides design.

Offer informative feedback: For every user action, there should be system feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial.

Design dialogs to yield closure: Sequences of actions should be organized into groups with a beginning, middle, and end. Informative feedback at the completion of a group of actions gives operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans from their minds, and a signal to prepare for the next group of actions.

Prevent errors: As much as possible, design the system such that users cannot make serious errors; for example, grayout menu items that are not appropriate and do not allow alphabetic characters in numeric entry fields. If a user makes an error, the interface should detect the errorand offer simple, constructive, and specific instructions for recovery.

Permit easy reversal of actions: As much as possible, actions should be reversible. This feature relieves anxiety, since the user knows that errors can be undone, thus encouraging exploration of unfamiliar options. The units of reversibility may be a single action, a data-entry task, or a complete group of actions, such as entry of a name and address block.

Support internal locus of control: Experienced operators strongly desire the sense that they are in charge of the interface and that the interface responds to their actions. Surprising interface actions, tedious sequences of data entries, inability to obtain or difficulty in obtaining necessary information, and inability to produce the action desired all build anxiety and dissatisfaction.

Reduce short-term memory load: The limitation of human information processing in short-term memory requires that displays be kept simple, multiple-page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

Prevent errors

The importance of error prevention (the fifth golden rule) is so strong that it deserves its own section. Users of cell phones, e-mail, spreadsheets, air traffic-control systems, and other interactive systems make mistakes far more frequently than might be expected.

One way to reduce the loss in productivity due to errors is to improve the error messages provided by the interface. Better error messages can raise success rates in repairing the errors, lowering future error rates, and increasing subjective satisfaction.

Improved error messages, however, are only helpful medicine. A more effective approach is to prevent the errors from occurring. This goal is more attainable than it may seem in many interfaces.

The first step is to understand the nature of errors. One perspective is that people make mistakes or "slips" that designers can help them to avoid by organizing screens and menus functionally, designing commands or menu choices to be distinctive, and making it difficult for users to take irreversible actions.

Integrating automation while preserving human control

The guidelines and principles described in the previous sections are often devoted to simplifying the users' tasks. Users can then avoid routine, tedious, and error-prone tasks and can concentrate on making critical decisions, coping with unexpected situations, and planning future actions.

The degree of automation increases over time as procedures become more standardized and the pressure for productivity grows.

With routine tasks, automation is desirable, since the potential for errors and the users' workload are reduced. However, even with increased automation, designers can still offer the predictable and controllable interfaces that users often prefer.

For example of the complexity of life-critical situations in air-traffic control emerges from an incident on a plane that had a fire on board. The controller cleared other traffic from the flight path and began to guide the plane in for a landing. The smoke was so thick that the pilot had trouble reading his instruments.

The goal of system design in many applications is to give operators sufficient information about current status and activities so that, when intervention is necessary, they have the knowledge and the capacity to perform correctly, even under partial failures.

Questions of integrating automation with human control also emerge in systems for home and office automation.

THEORIES

One goal for the discipline of human-computer interaction is to go beyond the specifics of guidelines and build on the breadth of principles to develop tested, reliable, and broadly useful theories.

Another way to group theories is according to motor-task performance (pointing with a mouse), perceptual activities (finding an item on a display), or cognitive aspects (planning the conversion of a boldfaced character to an italic one).

Critics raise two challenges:
Theories should be more central to research and practice: A good theory should guide researchers in understanding relationships between concepts and generalizing results. It should also guide practitioners when making design tradeoffs for products.
Theories should lead rather than lag behind practice: Critics remark that too often a theory is used to explain what has been produced by commercial product designers. A robust theory should predict or at least guide practitioners in designing new products.

levels of analysis theories
One approach to descriptive theory is to separate concepts according to levels. Such theories have been helpful in software engineering and network design. An appealing and easily

comprehensible model for interfaces is the four-level conceptual, semantic, syntactic, and lexical model.

- **The *conceptual level*** is the user's "mental model" of the interactive system. Two mental models for image creation are paint programs that manipulate pixels and drawing programs that operate on objects. Users of paint programs think in terms of sequences of actions on pixels and groups of pixels, while users of drawing programs apply operators to alter and group objects. Decisions about mental models affect each of the lower levels.
- **The *semantic level*** describes the meanings conveyed by the user's input and by the computer's output display. For example, deleting an object in a drawing program could be accomplished by undoing a recent action or by invoking a delete-object action. Either action should eliminate a single objectand leave the rest untouched.
- **The *syntactic level*** defines how the user actions that convey semantics are assembled into complete sentences that instruct the computer to perform certain tasks. For example, the delete-files action could be invoked by a multiple object selection, followed by a keystroke, followed by a confirmation.
- **The *lexical level*** deals device dependencies and with the precise mechanisms by which users specify the syntax (for example, a function key or a mouse double-click within 200 milliseconds).

This approach is convenient for designers because its top-down nature is easy to explain, matches the software architecture, and allows for useful modularity during design.

Designers are expected to move from conceptual to lexical and to record carefully the mappings between levels. This model was very effective in the early days of computing, when command-line input common and implementers had to low-level syntax lexical-analysis programs.

Stages-of-action models

Another approach to forming theories is to portray the stages of action that users go through in trying to use interactive products such as information appliances, office tools, and web interfaces.

1. Forming the goal
2. Forming the intention
3. Specifying the action
4. Executing the action
5. Perceiving the system state
6. Interpreting the system state
7. Evaluating the outcome

A stages-of-action model helps us to describe user exploration of an interface

As users try to accomplish their goals, there are four critical points where user failures can occur:

(1) users can form an inadequate goal,

(2) users might not find the correct interface object because of an incomprehensible label or icon,

(3) users may not know how to specify or execute a desired action, and

(4) users may receive inappropriate or misleading feedback.

Refinements of the stages-of-action model have been developed for other domains.

(l) recognize and accept an information problem,

(2) define and understand the problem,

(3) choose a search system,

(4) formulate a query,

(5) execute the search,

(6) examine the results,

(7) make relevance judgments,

(8) extract information, and

(9) reflect/iterate/stop

GOMS and the keystroke-level model

An influential group of theorists at Carnegie-Mellon University carried forward the idea of levels of analysis by decomposing user actions into small measurable steps.

They proposed two important models:

- The *goals, operators, methods, and selection rilles* (GOMS) model
- The *keystroke-level model*

The GOMS model postulated that users begin by formulating goals (edit document) and subgoals (insert word). Then users think in terms of operators, are "elementary perceptual, motor, or cognitive acts, whose execution is necessary to change any aspect of the user's mental state or to affect the task environment.

The keystroke-level model is a simplified version of GOMS that predicts times for error-free expert performance of tasks by summing up the times for keystroking, pointing, homing, drawing, thinking, and waiting for the system to respond.

GOMS works nicely for describing steps in decision making while carrying out interaction tasks, such as text editing in a manuscript.

Extending GOMS 'with if then rules to describe the conditions and actions in an interactive text editor proved to be a powerful addition. The number and complexity of production rules gave accurate predictions of learning and performance times for five text-editing operations: insert, delete, copy, move, and transpose.

Consistency through grammars

An important goal for designers is a *consistent* user interface. However, the definition of consistency is elusive and has multiple levels that are sometimes in conflict; it is also sometimes advantageous to be inconsistent.

The argument for consistency is that a command language or set of actions should be orderly, predictable, describable by a few rules, and therefore easy to learn and retain.

Consistent	Inconsistent A
delete/insert table	delete/insert table
delete/insert column	remove/add column
delete/insert row	destroy/ create row
delete/insert border	erase/draw border