DESIGNING CONSUMER PRODUCTS FOR EASE OF USE

Whether the task is to design a better display, phone, or telecommunications system, the job of the human factors engineer is to ensure that the product has the right functionality, is easy to learn, and is easy to use. Although there are a number of guidelines for user interface design, their value is limited when designing consumer products in which the competing interests of cost, marketability, consistency, entertainment, and usability must all be balanced in a case-by-case basis. Therefore, rather than provide a catalog of design recommendations or a review of the technologies themselves, the objective of this article is to sensitize the reader to some of the basic human factors issues in the design of consumer products, and the methods human factors engineers follow in arriving at design solutions. Although specific guidelines are not referenced, a short list of excellent general resources for the user interface designer is provided at the end of the article. Throughout this article the example of a telephone as a typical consumer electronics device will be used, but the issues apply to all products that require both human input and system output.

HUMAN FACTORS METHODS

User-Centered Design

The guiding philosophy of user-centered design for consumer electronics is simple: Product designers need to include consumer input at *all* stages of design, including the earliest conceptual stages. It is important that consumer product designers realize that, as individuals, they are usually not representative of typical consumer users and should avoid the natural temptation to design just for themselves. Data from representative consumer users must be included in the user-centered design process from early analysis of user tasks to final empirical usability testing of the completed product, including its customer manuals and any associated training. Synonyms often used to describe user-centered design include: human factors engineering, ergonomics, and usability engineering. In this section will be defined the principles of human factors and some of the more common methodologies that can be applied to consumer electronics product design.

Benefits of Human Factors/Usability Engineering. Human factors engineering is the application of behavioral sciences and applied psychological principles and methods to the design of the user interfaces to products. Human factors specialists work most effectively as full participating members of product design teams, where they can apply their expertise and unique perspective as “advocates for the user” to all team problems. The goal is to design user interfaces that

- Are attractive and inviting, leading to increased sales
- Are easy to learn and easy to use because they are consistent with both general user expectations and specific ones about how particular products should work
- Are consistent with user interface standards, including de facto standards
- Help maximize user productivity by minimizing user error, task completion time, and user anxiety
- Match consumer capabilities and limitations
- Reduce product development time and cost because user interface rework is minimized, and because they can easily adapt to next-generation user interface features and functionality
- Are reliable and supported by useful and usable help, minimizing repairs and returns

Human factors expertise comes from both specialized training and experience in knowing how to reliably measure human behavior. To be most effective, the human factors team member must be knowledgeable about product technology and architecture, as well as being expert in applying the principles of human factors.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*.

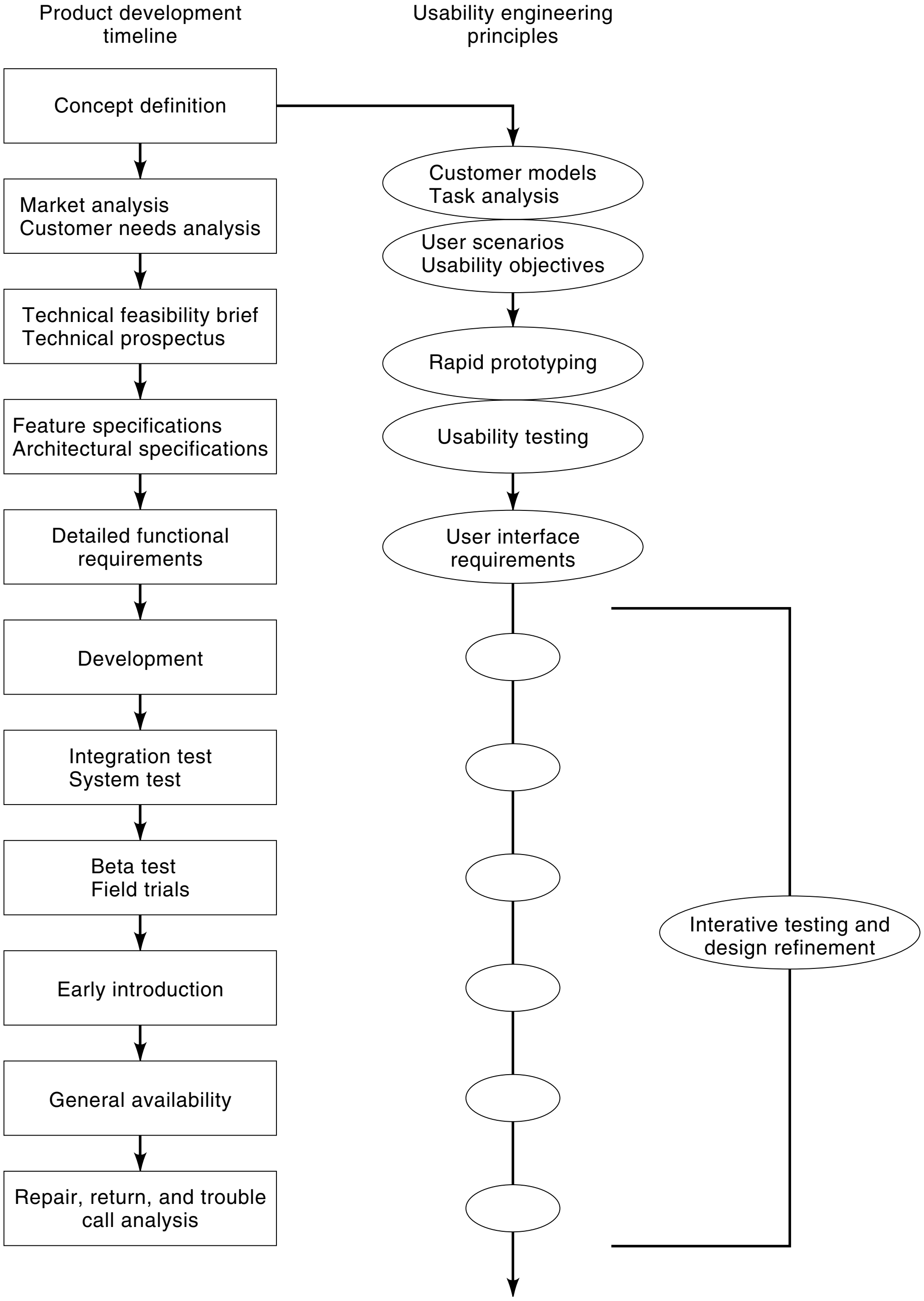

Figure 1. Usability engineering principles in the product realization process. Note that up-front human factors design work should be completed before product development begins. Guidelines and standards, as well as data from human factors studies, are used to build both the customer models and prototypes.

Human factors should be involved in most phases of the product realization process. Figure 1 shows how core usability engineering principles can be incorporated into the product realization process. This involvement is critical in each phase of product realization from initial concept development, through design, test, and implementation.

Core Usability Engineering Principles

A subset of user-centered design activities are considered core usability engineering tasks. Human factors specialists either carry out these tasks themselves or assist and guide other project team members in their implementation.

Concept Development and Product Planning

Customer Profiles/Customer Models. These detailed descriptions of typical consumer product configurations include descriptions or profiles of different classes of users and the features they would use. Interactions of features and their frequency of use can also be specified. These profiles answer the basic question *who are the users* and *what are their goals?*

User Scenarios. These are detailed descriptions of the actual sequence of tasks users would follow in accomplishing a goal using the product as described in the customer model. User scenarios are derived from detailed task analysis data obtained from comprehensive studies of consumers and the activities and tasks they want to accomplish using the product. These user scenarios answer the basic question *how will the users use the product?*

Usability Objectives. These are measurable and agreed upon criteria for describing a typical user's task performance (speed, accuracy, and productivity) and satisfaction (survey ratings) when completing the user scenarios that were derived from the customer models. A typical usability objective might be worded as follows: Eighty percent of first-time users without the aid of instructions should be able to put a call on hold, answer a second call, and return to the first without error. Ninety-five percent should be able to perform the task after reviewing the instructions for five minutes. Usability objectives address the basic question *how easily can consumers use the product?*

User Interface Design and Testing

Rapid Prototyping. New software tools allow user interface prototypes, solid models, or simulations that mimic the user interface to be developed early in the development process. A good prototype can be easily modified to allow rapid iterative design after testing with representative users. Prototypes range from low-tech pencil mockups to high-tech computer simulations or physical models. Sometimes these prototypes, when refined, are incorporated directly into the final product code or plastic.

Usability Testing. Evidence for the suitability of the prototype is determined through careful testing. This testing ranges from simply observing subjects using the prototypes to intensive monitoring of all the user's interactions, including capturing every button press and even eye movement. The subjects, matching the customer models, are tested on the customer scenarios to determine if the usability objectives are met. The more closely the test environment mimics the real environment in which the product will be used, the more confident one can be of the validity of the test results.

Variables that are thought to have significant influence on user interface quality can be systematically controlled and varied, and their impact on user performance and satisfaction can be reliably measured. Usability testing is best carried out in two phases of design: (1) early in the design cycle using simulations before final requirements are written, and (2) later during system test of the extended product, including documentation and any user training. Testing and design are most often an iterative process. A round of testing reveals problems. Designs are reworked and new testing is performed.

Other Types of Human Factors Evaluations

Described above were the core human factors principles of customer models, customer scenarios, usability objectives, prototyping, and usability testing. Testing is central to the work of the human factors engineer and it is the most important contribution he or she brings to the development effort. In addition to controlled laboratory testing (often with paid subjects), other types of evaluations are frequently performed.

Literature Reviews. Human factors engineers often apply results of *previous* human factors to new designs. These studies come from a large number of scientific research journals in applied psychology, cognitive science, marketing research, management science, and other behavioral science disciplines. Meta-analysis is a related literature review methodology for statistically analyzing a group of studies where each individual study is treated as one of many data points.

Peer Reviews. This method allows the logical flow and potential flaws in the user interface to be examined in formal round table reviews and walkthroughs of paper designs. These reviews could include designers and potentially representative users methodically stepping through all details of user scenarios and in a step-by-step manner uncovering usability defects.

Expert Analyses or Heuristic Evaluations. Some aspects of the user interface are best tested by trained evaluators. For example, voice quality can be evaluated by trained listeners who understand the various sorts of impairments typically found by telecommunications devices. After a few hours of listening, they can provide specific recommendations for improvements.

Field Studies. Data from actual users can be collected in field settings with or without control of independent variables that are thought to affect user performance or satisfaction. Data can be collected through surveys, structured interviews, observation, videotape, software capture, electronic instruments, transaction logs, user diaries, and so forth. Usability testing against usability objectives can also be obtained in field settings if careful controls are implemented.

Focus Group Sessions. User interface design issues can be examined quickly by having human factors specialists, as part of a project team, design and conduct focus group interviews with representative users. In a focus group, 8 to 12 users discuss, in a group interview setting, a specific topic such as alternative user interfaces or priority of product features needed. The discussion is controlled and focused by a professional moderator. Sessions are often videotaped and viewed live through one-way glass mirror windows. Members of these focus groups can also be recruited from customer panels used by marketing researchers.

Participatory Design. Representative users can assist in producing user interface mock-ups in a small group setting moderated by a human factors specialist. Marketing and development specialists will often participate. The mock-ups are often only made from index cards, notes, and other low-technology devices.

THE HUMAN–MACHINE DIALOG

A person using an electronic device is participating in a *dialog.* The initial task of the user is to understand the machine's state and to communicate his or her desires, "I would like to place a call." "I would like to set the alarm." Today's technology can provide the user a variety of ways in which to communicate; pressing buttons, speaking, pointing, writing, or even just looking (with eye-tracking equipment). Similarly, the machine can respond with music, speech, text, images, or even tactile stimuli. However, in consumer devices, the user typically communicates by pressing keys, while the device responds with simple auditory and visual displays.

In the next section some of the human factors issues in the design of human input to a system will be examined, and in the following section attention will be given to the system output.Three points should be emphasized:

1. It is convenient to conceptually divide the human–machine dialog into input and output; however, one can

no more design one half of this dialog independently of the other than one can dance a tango alone.

2. Although a number of "rules of thumb" are mentioned, the human–machine dialog must always be tested.
3. Even with a good understanding of the technology, the design of the human–machine dialog must begin with a full understanding of three fundamental components on the "human side" of the human–machine dialog: the users (What will their expectations be? What other devices will they be familiar with? What perceptual or motor disabilities might they have? How sensitive will they be to cost, aesthetics, and performance?); the tasks (What tasks will be performed and in what order? What information will the device need to communicate and when? What will the user's goals be? What other ways might the user consider to perform these same tasks?); and the environment (Light or dark? Loud or quiet? Will the device be in arm's reach?).

The best user interfaces arise out of careful up-front analyses, frequent testing, and an appreciation that the success of the human–machine dialog is determined by the complex interaction of *all* the components of the human–machine system, from the simple feedback tone to the cultural biases of the user.

THE HUMAN–MACHINE DIALOG: USER INPUT

Although it is useful to view the user's interaction with a machine as a dialog, this human–machine dialog is in many ways an unnatural one. Typically, dialogs are carried on with fellow humans using sight and sound—gestures and speech. However, the most common method of communicating with a machine is by touch—pressing buttons, flipping switches, and pulling levers. Communication by touch requires learning new rules of interaction, rules that are often arbitrary. For example, we tell the plumbing system what we want by turning a knob, following a mapping of input to output that is often unpredictable. We tell our car what we want by a partly standardized and partly unique means of knobs, dials, and pedals. We tell our microwave what we want by pressing buttons, following sequences that vary even within manufacturers. We tell our word processor what we want by pressing an uncountable variety of key combinations. Although ubiquitous, communication by touch is often a source of frustration.

On the other hand, interaction by touch can be quite serviceable. Although the mapping between a command and the desired result may be arbitrary and hard to learn, once learned, there is little or no ambiguity. One either presses the HOLD key or one does not. Humans can become fairly accomplished at pressing keys. A good typist can enter 100 words a minute using a standard QWERTY keyboard (named for the first six letters on the top row of keys) and a trained stenographer can enter perhaps twice as many words a minute using a chord keyboard.

To clarify the nature of this interaction and to determine ways of making it less difficult for the user, it is convenient to approach the human–machine dialog at three levels: the physical (e.g., keys, displays, speakers), the semantic, (e.g., labels, error messages), and the logical or syntactic (e.g., menu organization, feature interaction). The logical level can best be understood in the context of a specific application, and so will not be discussed here. Several design "challenges" concerning the use of touch input will be reviewed, challenges of which engineers need to be cognizant. These issues cluster around the fundamental issues of size, cost and reliability, action, and intuitiveness. The following sections will discuss many of the input and output issues (see Table 1) that have been the subject of human factors research.

Design Challenges: Physical Level

Key Size. Keys take up space. Keyboards and keypads are quickly becoming the gating factor in the attempt to reduce the size of consumer products from cellular phones to laptop computers. The ideal key has a 12 mm to 15 mm diameter with 18 mm to 19 mm center-to-center key spacing between neighbors. If the key size or key spacing is any smaller or

Table 1. Some Areas of Human Factors Research on the Human–Machine Dialog

USER INPUT
Keys and Switches
size, spacing, shape, surface, travel, force, key de-bouncing, protection against unintentional presses
color, layout, grouping, labeling, coding, response mapping, feedback (sound, volume, delay)
Pointers
type, technology, resolution, select activation
cursor (size, shape, blink rate), cursor repositioning
Voice Recognition
keyword sets, vocabulary size, verbal and tone prompts, feedback, error handling
activation/deactivation, speaker independent/dependent, key input alternatives
SYSTEM OUTPUT
Visual Displays
size, resolution, flicker, parallax, visual angles, brightness, color, contrast, placement
icons (use and design of), scales (marking, coding, intervals), analog versus digital representations
text displays (line spacing, line length, character size and fonts, abbreviations), scrolling methods
coding schemes, coordination with user input, content, stimulus set
Auditory Displays
frequency, timbre, rhythm, loudness, duration
coding schemes, coordination with user input, content, stimulus set

larger, performance suffers (Bullinger, Kern, and Braun, 1997). To take a specific example, a basic mobile phone with 24 functions (send, end, power, redial, volume keys, etc.) and a 12-digit keypad needs at least a surface area of 160 cm^2. The surface area of a cellular phone available for keys is approximately 60 cm^2. One can see why buttons have become smaller, interkey spacing tiny, and multiple functions are loaded onto single keys. In addition, it is one more reason to pursue voice input, given that a mobile phone already has a microphone and speaker.

Cost, Reliability, and Harsh Environments. Touch-input devices cost money. The more reliable the touch-input device the more expensive. Keys, switches, and dials involve separate parts and processing in manufacturing, which is one reason manufacturers prefer that designers use standard sizes, shapes, and orientations in their designs. Because physical contact is made between the input device and the user, labels must be durable—another source of cost. Moreover, because the physical contact often occurs in environments that rarely match that of a "clean room," keys and other physical input devices can admit dust and grime, causing keys to malfunction. Consumer products often must pass stringent humidity and dust performance tests. The ideal consumer item, in terms of reliability, is a sealed, insulated box. All products aspire to this goal; however, keys admit openings that introduce dust, water vapor and, worst of all, electric static discharge (ESD).

Key Feedback. Human-to-human dialogs are maintained by a complete set of subtle feedback loops of eye gazes, nods, sounds, and gestures. A machine needs to provide the human with a similar set of confirmations, called *auditory, visual,* and *tactile feedback.* (Only the immediate physical confirmation of a user action are discussed here. In addition, the user interface of the product should inform the user of the appropriateness of the user's action and the time it will take to satisfy the request. This is most often done by playing "happy" and "sad" tones, by displaying an indication that the command is being processed, by displaying the expected time it will take to execute the command, or by displaying an error message.)

All buttons require auditory feedback, often called *key click.* The click should be loud enough to hear, but not so loud as to be annoying. Most often this feedback is provided by the key movement itself, but if this is not possible, then it should be artificially generated within 100 ms of pressing the key.

Visual feedback is often provided by the switch or button physically moving. This is usually sufficient, but additional cues are often easy to add. Many switches can reveal a colored surface in one position, immediately announcing its state. In some cases, when the response or state of the machine is particularly important, for example, on a power strip or kitchen stove, a light-emitting diode (LED) or other visual display indicator is associated with the position of the switch.

Key Travel. In order to reduce the cost of keypads, designers in a number of products have tried "flat" keys, also called *membrane keys* or *no-travel* keys. These keys have no or little *travel* or *displacement,* that is, the keys do not move up and down. They can be manufactured as a single plastic sheet over a matrix of contacts, solving the problems caused by cutting openings into the housing, as discussed above. No-travel keys have the advantage of being much easier to manufacture and to keep clean. In cases in which key presses are relatively infrequent and in which cleanliness is a concern, for example, a microwave oven, flat keys may make sense. The main problem with membrane dial pads is that users don't like the lack of tactile feedback. Numerous attempts to solve the problem of feedback and key location have not succeeded in making them acceptable. Users prefer about 3 mm to 4 mm of travel. As with key size and spacing, more or less will result in loss of preference and performance, in particular with rapid key sequences, as happens in dialing or typing. It should be noted as well that although long fingernails present a problem to any type of key input, it is especially a concern with flat keypads.

The feel of the key is more than the distance the key travels. A key offers resistance to the finger that changes through the course of the key travel. These *key force curves* can be quite complicated. For example, a key can exhibit low resistance initially, gradually increasing in resistance until it reaches a maximum, or it can do the opposite. Most typewriter keys reach a point of high resistance in the middle of the downward movement, after which there is a release, marked by a click.

Key Force. How much force overall should be required to activate a switch? It depends how the key is used, but two principles are widely accepted today. A force of 50 g to 100 g is desirable for keyboards and keypads. Key forces of 200 g to 300 g are more appropriate for keys that are not pressed in rapid sequence. Lower forces result in too many accidental key presses and greater forces result in fatigue and missed keys. Second, the keys of a keypad should not vary in activation force from key to key by more than 5% to 10%. Fortunately, the manufacturers of switches have studied these phenomena and have designed switches to satisfy subtle preferences.

Special Considerations. When keys might be pressed in the dark or while looking away from the keypad, users should be able to orient their hands by touch. Position of the keys or switches in itself will help orient the user, of course, but additional cues can be provided by adding *nibs* molded into the upper key surface (e.g., on the J and F key of the keyboard or the 5 key on the dial pad), or by varying the *shapes* or *surfaces texture* of the keys.

Devices intended to be operated with one hand present new challenges. Keys that are frequently used should be accessible by the arc made by the thumb on the hand holding the device. The side faces of the device can frequently be accessed by the index finger. However, the designer should be aware that the physical accessibility of keys will change for left- and right-handed users.

Design Challenges: Semantic Level

In addition to having controls that are *physically* "right-sized," the user needs to know which control is the right one at the right time. To support the user in this *semantic* task, the designer needs to consider the key arrangements and coding schemes used to assign *meaning* to keys and key action.

Key Layout

General Guidelines. There are few solid guidelines to help with the arrangement (or *layout*) of keys or switches on a device. Certainly there are standards for well-known input devices like the telephone keypad and the typewriter keyboard, but when designing new arrangements there is little to go by. Nevertheless, a few principles are generally followed: HELP functions should be placed on the upper left, EXIT functions should be on the lower right, SHIFT functions should be on the lower border. So called "dangerous functions" (functions that take time to undo, e.g., a POWER ON/OFF switch or a RECORD button) should be hard to press by accident. Such switches can be placed on a surface that is physically hard to reach, the switch can be recessed or covered, it can be made very small, it can require high activation force, two-finger, or multiple key presses. On the syntactic level of input design, a *confirmation prompt* (e.g., "Are you sure?") can be included in the dialog.

Grouping. One of the most successful ways of guiding the user in the human–machine dialog is to group keys on the control pad by the use of physical spacing, color, position, or graphics. On a telephone, the 0 to 9 key pad is typically grouped in a 3 by 4 array, with all 12 keys the same color and size. The HOLD key may be visually separated by its color and shape. Rules for keypad grouping include: like functions should be clustered; keys should be arranged from left to right in the order in which they will be used; the most frequently used and the most important keys should be placed on the outer edge of a keypad; and keys should be physically close to the any visual feedback that is tied to the key. Key positioning, sizing, and labeling is both an art and a science. Frequently the only way to produce a successful design is by prototyping and testing.

Key Coding. There are many ways to code or label a key: size, shape, position, texture, color, or even its action or operation. The most common method of coding is with a text label; however, often multiple coding schemes are used. For example, the reset button on many devices is smaller, colored, recessed, and labeled "reset." Position can become a coding scheme as well. The space bar is not labeled on most keyboards. If it is changed in position, it will lose its meaning. The shape of the key itself can serve to code the function, as with arrow-shaped cursor keys.

General Concerns. There are a number of concerns the designer needs to be aware of when selecting a method of key coding. Will the hand obscure the label? (In general, it is preferred to place the label above a key for this reason.) Will the user have to look at the control panel to operate the key? Will a user with a color perception limitation be able to operate the device?

Labels. Labeling a key may seem easy, but because of the small space typically available, and because of the need to make the label both visible and meaningful, it is often a challenge. Certainly keys like HOLD or TAB are well understood in the English-speaking world, but often what the device wants to "say" to the user in a dialog is much more complex, for example, "Press me in order to reach your voice mail system," or "I can help you place a conference call, but first you must establish a call, then press HOLD, then press me, then dial the next party. . ." In these two examples, MAIL and CONF may have to do. In some cases, the designer does not even try to label a key, leaving it up to experience. (What would an acceptable label be for the left or right mouse button? And would the text label help anyway?) In other cases, terms are invented with the hope that users will learn their assigned meaning, like FLASH. Small space contributes to the problem of how to label other types of input devices as well. How does one indicate the maximum and minimum volume on a slide switch, or the treble and bass on a knob? *Symbols* can be useful. However, the danger of symbols (or icons) is that, although they may seem clear to the author, they may not to the user. If one needs a key to say "This is the key you press to see the next page of your directory," an → key might seem obvious. But does it mean "next page," "next line," "next character," "change my clock forward," or perhaps "print an arrow?" In some cases, the mapping becomes completely arbitrary and, therefore, must be learned as with the ◇ key on some public phones, signifying "Press this key to place a new call without having to reenter your billing number." (On some other public phones this key is labeled NEW CALL.)

Language. For products that are to be used by consumers who speak different languages, new issues are introduced. The owner's manual can be translated, but is there room to put multiple labels on the faceplate? Products to be sold in the international marketplace often rely on icons, for example, the commonplace icons for "rewind," "stop," "record," on VCRs. These may seem intuitive once learned, but they still need to be learned. For this reason, icons must be used with care, and preferably should be accompanied by text.

Input-to-Output Mapping. Most keys are "digital" in nature. They are either up or down, on or off. They work well for "digital" actions, like dialing a telephone number or muting a microphone. However, many actions are viewed as analog by users, for example, controlling volume, brightness, tone, temperature, and even finding one's place in a long text or tape. These sorts of analog controls are best served by analog input devices, like dials and slide switches. Dials and slides also have the advantage of "visually holding" their setting, that is, the user can determine the current setting by looking at the input device. However, analog controls are almost always more expensive than keys, and they are more prone to reliability problems over time. Therefore, they are less preferred by those who want to minimize the cost of the product.

Improving the Human–Machine Dialog by Simplifying the User Input

From telephones to clock radios, the trend in many consumer electronics devices is to put more functionality into a smaller package. These two conflicting directions demand difficult trade-offs among the various guidelines for good user interface design, with the solutions typically requiring smaller keys, eliminating functions, doubling-up on keys, or looking for new ways to provide user input in the human–machine dialog.

Reducing the Size of Keys. More keys on the same surface area means smaller keys. Some products, for example pocket calculators and digital watches, with a "self-selected" user population, have taken this approach of shrinking keys. How small can keys get? There seems to be no limit. Some watches require the user to use a stylus to use the functions. To optimally use the available area, the designer can make some more frequently used keys larger than other keys, for example on a digital voice recorder, the PLAY and RECORD keys are often larger than the other keys. For serious typing, however, users will not accept less than the standard keyboard, and for occasional typing three-quarter size keys seems to be the minimally acceptable keyboard. For most applications, users are not willing to accept on-screen keyboards for any amount of data entry.

Multifunctional Keys. A more complex design issue is the trade-off between more keys, with each key assigned to a single function, and fewer keys, with each key assigned multiple functions. To take a simple example, a phone could have separate volume control keys for the speaker, the ringer, and the feedback tones, or the phone could have a single pair of up and down keys, which adjust the *speaker* when the user is on a call, the *alerter* when the phone is ringing, and the *feedback tones* otherwise. To take it one step further, a single pair of keys on a cordless answering machine could be used to adjust the volume, the channel, and the message number—three very distinct functions. Doubling-up can be a problem for users because they have to be cognizant of the state of the device, although certainly we have all become comfortable with state found on a standard keyboard, namely, pressing the 4/$ key prints a 4 unless the keyboard is in the “Shift State,” in which case it prints a dollar sign.

It should be noted that consumers have objected to the overuse of highly state-dependent input devices with long lists of “shift,” “ctrl” and “alt” combinations. Telephone service designers build complex services using only the 10 digits, star, and pound on the common telephone keypad. It is not only a question of asking the user to remember more; it also results in underutilized functions. It is well known that replacing a “star-function” (e.g., *72) with a button or menu choice results in a dramatic increase in the usage of that service.

Making the current state determine the function of a key is confusing to many users, except in well-understood situations. It requires more learning and can often lead to unexpected actions in the human–machine dialog. In particular, casual users and novices seem to have more difficulty with state-dependent input devices.

Soft Keys. The third method of handling the problem of too many keys is to do away with them entirely, for example, by putting the keys on a display device. The user requests the function either by pressing the screen directly, using a *touch screen,* or by pressing an unlabeled key that is associated with a particular location on the screen, often called a *bezel key* or a *soft key.*

The clear advantage with touch screen keys and soft keys is that only those keys that are relevant in that particular state need to be visible. For example, when not on a call there is no need for a HOLD or CONFERENCE function. Similarly, when on a call there is no need for a REDIAL or PHONE LOCK function. The labels can dynamically change, depending upon the state of the device. Not only can one reduce the number of keys, but the designer can *focus* the user’s attention on a particular function. For example, if there is an incoming message, the MAIL label/key can flash on the screen.

The disadvantage of touch screen keys is that users do not like to hold their hands up to their screens. In many applications, fingers are too wide, requiring the use of a stylus. Moreover, when fingers are used, the displays quickly become dirty. This last fact cancels any advantage of touch screens in many applications. This is not a problem with bezel keys, of course.

Another concern is that users do not have the comfortable feeling of consistency in the user interface. Labels appear and disappear. Typically, in soft key interfaces, the user scrolls through all the various functions, potentially leading to the feeling of being “lost in menu space.” Pressing a particular key does not always have the same result and not all functions are visible at all times. There is some anecdotal evidence that both aspects of soft-key interfaces can cause both younger and older users difficulty with the human–machine dialog.

Pointers

Pointers can be viewed as another type of soft key device because, using a pointer, the user can select functions displayed on a screen. (Pointers are more versatile than other soft keys because they also permit continuous movement, which greatly facilitates activities like dragging objects and selecting areas on a screen.) A pointing device is really the coordination of three components: an onscreen cursor, a means of moving the cursor, and a select key. (Pointers may include additional functionality.)

The three dominant classes of pointing devices are the mouse, the trackball, and the tablet. Each has its own advantages and disadvantages. When selecting the appropriate pointing device, the designer must consider the work area, the capabilities of the user, the application, and performance requirements. For example, older users tend to prefer trackballs over mouses, probably because trackballs are more accurate, although slower (Hoag, 1996). The positioning function and select function are most closely integrated in the mouse, allowing for faster positioning and selecting, but causing more errors in users without fine motor control. With trackballs the cursor can remain in motion after been initially set in motion, an advantage for some games. Tablets are preferred for graphics work because of the similarity of movement with pens and brush and because the cursor can be *repositioned* more easily than with mouses or trackballs. Tablets also permit handwriting-recognition applications. Trackballs and tablets also do not require as much desk area and can be more successfully integrated into hand-held devices. All in all, mouses have proven to be the most preferred pointing device for general-purpose computer applications.

Voice Input

Yet another way to respond to the challenge of more functionality in a smaller space, and to provide more a more intuitive dialog at the same time, is voice input. Recent years have seen a greater number of household or home-office devices sold to general consumers that utilize speech recognition technology. Examples are VCR remote controls that understand scheduling commands and portable “personal organizers” that record and play messages using voice command control. There is no doubt that speech recognition technology has advanced tremendously, but it should be noted that it is common to find that the level of technology which can be placed in portable device or inexpensive home products will be less advanced than what can be found, for example, in office computer systems and in the telephone network.

The most fundamental choice to be made in speech recognition is whether or not using speech recognition will be perceived as valuable or useful to users of the device. Here it is essential to recall some generalizations about the technology:

- Despite the hyperbole which often accompanies advocates of speech recognition, "speech" is not, by its very nature, necessarily a natural way of interacting with a machine. Using the auditory/verbal channel may be wise in some circumstances and not others.
- Regardless of the validity of the "naturalness" argument, even the best technology does not approximate natural speech interaction. When the general public loosely speaks of "natural conversation" it is really referring not only to speech recognition (the ability to identify spoken words), but also to natural language understanding (using background knowledge and inferences about the real world to hold conversations). Technology to do this in consumer products is in its infancy, and can only, at best, mimic conversation through clever design.
- Even the best speech recognition technology does not recognize words with 100% accuracy, and often, in the field, does a much poorer job than a human being. Thus, the design of any system must recognize and expect users to deal with failures of the system to recognize their speech correctly. Nevertheless, there are circumstances in which speech recognition may be very useful to users, namely, in situations where
  - The user needs to have their hands free to do other things, for example dialing a cellular telephone while in an automobile
  - The user cannot use other input such as typing, for example disabled users with motor impairments use voice recognition to control their personal computers
  - The system is complex and/or the task is difficult enough that the user's work load is high; in such cases, speech input can be more accurate than manual input (such a situation, however, is more likely to occur in cockpits rather than consumer products.)
  - The device or system needs to be controlled over a distance, and a remote control is not available or problematic in some other way.

In addition, speech recognition may end up simplifying some controls because a series of keypress commands, can be consolidated into a single voice command. (Although, in such cases, it may be quite possible to design such consolidation in manual control without resorting to voice control.)

The most important rule is that speech recognition should be used for noncritical tasks because of its error-prone nature. If, however, errors can be tolerated, and recovery is easy, then speech may show some evident advantages. Even with a task like keying in numbers on a calculator or dialing a telephone, where research has shown that manual input is more efficient and accurate, voice input for such tasks can be very useful in, for example, speed dialing or in "hands-busy" situations like car telephones.

Once the decision is made to use speech recognition, careful attention must be paid to design. Seemingly subtle differences in prompt wording and vocabulary choice can have significant effects on recognition performance. Some general principles in designing a voice recognition user interface are:

1. The most crucial design element is feedback and error recovery. Always give feedback as to the results of a recognition. This may consist of repeating back the recognized word or asking for a confirmation. In many cases it is better to begin executing the next step in the procedure with opportunity for canceling. In any case, the results of a recognition should be immediately clear to the user.
2. Always make it easy to cancel an ongoing action which may be the result of a misrecognized word.
3. If error feedback is spoken, do not word the prompt in a way that blames the user. Avoid an infinite loop of error and reprompting. If possible, suggest another method of accomplishing the task.
4. If the system is performing outside of its operating environment, then it should inform the user in some fashion (e.g., too much environmental noise, or pauses too short between words). Principally, informative feedback in response to errors might accomplish this, or status displays. (On the other hand, excessive detail in error messages is not good either. Transactions should be quick; otherwise, use of the device can be too tedious. Also, the system may not always be very accurate in determining the reason for a recognition failure.)
5. Because speech recognition systems can fail utterly for some people, or even for momentary periods of time for users who normally do not have problems (e.g., due to background noise, colds, stress), an alternative method of completing the task should be available if possible. Often, this means that the device should be able to complete all its functions by key presses as well as by voice.
6. Use a vocabulary that is familiar to users and that is consistent with the common vocabulary of the task being done. The vocabulary should be easy to remember and it should be consistent, regardless of the mode the device is in or the task being accomplished. Command lists should be consistent with vocabulary used in documentation or any other visual materials (e.g., labels on the device).
7. Generally, it is recommended that users not be burdened with creating the vocabulary for an application. A vocabulary should be created. In addition to the added effort, there is evidence that users have trouble remembering the words that they chose. There are clearly exceptions to this rule, as in voice-dialing applications, where users should be allowed to create words that speed-dial telephone numbers, such as "Joel's office," "Mom at home," and so on.
8. Be aware of natural speech habits and synonyms. For example, users will more likely say "oh" than "zero" for the number zero (even when instructed otherwise), so it is imperative to build-in recognition of "oh" as a speech synonym for 0. Alternative pronunciations of words should also be accommodated. If there is a general overall rule, it is that the design of the voice-recognition

> user interface should allow users to be focused on the task they want to do, not on the mechanics of the interface.

Well-designed user input alone does not ensure a successful user interface, of course. Attention will now be turned to the device's contribution to the human–machine dialog: visual and auditory output.

THE HUMAN–MACHINE DIALOG: VISUAL AND AUDITORY OUTPUT

As noted in the previous sections, on most consumer communication devices, users initiate and maintain their side of a dialog by manipulating switches (e.g., keys) on most consumer communication devices. The device communicates its capabilities, responses, and requests via *displays* (the term is used to refer to both visual and auditory displays).

In many systems designed before the advent of the microprocessor, displays were not necessary because the state of the device was immediately observable. For example, when a handset is out of its cradle the user knows there is an open circuit. When the tube glowed, the radio was on. Compare these cases with a speakerphone, in which an LED or message on a display needs to convey the state of the device to the user. In such a situation, the designer must deal with symbolic and often arbitrary "languages" to communicate with the user. Does a lit LED next to "battery" label mean that the battery is being charged or is already charged? How bright and what color should the LED be? Should it blink? At what rate? The color, intensity, flash rate, and position of an indicator are part of the device's language.

The same can be said of auditory feedback that was intrinsic to the operation of a machine, compared with symbolic auditory displays. When one puts a quarter into a coin phone, one can hear it fall down the chute. If the quarter is returned, one can hear it land in the coin-return bucket. On the other hand, how does one know whether the system accepted one's credit card number? What sound does a credit card make when it is rejected? Today's electronic systems rely upon a constructed and coded "language."

Visual displays comes in four levels of increasing complexity. *Indicators* are often binary, or at least indicate one of a small set of states, for example, On, Off, and On-Hold. *Meters* indicate a single quantity, usually numeric, for example, altitude or battery charge. *Alphanumeric displays* display text and can be used to indicate to the user error states, prompts, tables, and help. *Graphic displays* are still more versatile, allowing the designer to display varied fonts, illustrations, and images. Auditory displays have roughly the same levels of complexity. Simple *indicators,* often called feedback tones, distinguish among a few states, for example, an incoming call on line 1, line 2, or the intercom. More complex auditory indicators can communicate a value, for example, the dialed number. More recently, designers have at their disposal auditory displays that can provide prompts, error messages, status reports, and even high-fidelity music.

Based on a human factors analysis, the designer selects the most appropriate type of display (e.g., visual or auditory, analog or digital), the implementation (e.g., size, color, loudness, frequency), the content (e.g., the words displayed or spoken), and extrinsic factors (e.g., the position of a display relative to the associated user input).

In this section, some of the human factors issues in designing the device's side of the dialog will be discussed. As when discussing methods of input, the concern is not technological, but rather psychological. How can the user's ability to detect and decode the device's communications be maximized?

The Four Basic Components of a Human–Machine Dialog

The device communicates to the user via a set of messages. The four ingredients in a successful communication are the same for auditory and visual dialogs. First, there must be a clear *need* to communicate; second, the message must be *detected* by the user; third, the message must be *discriminated* (from among the set of alternative messages); and fourth, it must be *interpreted.*

User Needs. The designer should be aware that, like all aspects of the user interface, what is a signal to one user, might be noise to another. One user might be interested in knowing that the cordless phone just switched to another channel. Another might not care. All users should not have to pay for the needs of a few. The designer should also remember that a message from a device is typically an indication that something has happened or is happening. In particular, a display that changes invites attention. Steady states should be represented by a steady display, or none at all. Gradations on scales should not be so fine grained that insignificant changes in state are marked by large changes in the display. This can especially be a problem when displays are located in the visual periphery, where the eyes are more sensitive to brightness changes and motion. A bright, digital desktop device can be distracting every time a digit changes.

Detection. First, the user must be able to detect (to see or hear) the message. It must be bright or loud enough. The ability to detect the stimulus, more than any other step in the process, depends upon the user's sensory capabilities and the environment in which the device will be used. Because of this, the device settings that most directly determine detection should be adjustable.

Discrimination. Detection is a function of the environment, the stimulus, and the capabilities of the viewer. Discrimination is determining which stimulus, among all the possible stimuli, was presented and therefore is a function of the *stimulus set.* To take a simple example, the task of discriminating between a low tone and a high tone, or a steady-on LED and a flashing LED, is an easy one for users. In each case, the stimulus set contains two stimuli. However, users find it very difficult to discriminate among ten different frequencies or flash rates. If the tones are assembled into a melody, or if the LEDs form letters and words, a large number of stimuli can be reliably discriminated. The need to build an easily discriminated stimulus still applies when one is using words. If one's command set includes the commands "delete," "cut," and "erase," and if each command is assigned a different function, there are certain to be misunderstandings in the human–machine communication.

When designing the stimulus (or message) set, it is not only important that each message, viewed individually, is the "right one," but also that the individual messages are easily discriminated from their mates. *Multidimensional* stimuli (e.g., tones that vary in frequency, duration, and loudness) may be required. Extensive testing is recommended to come up with an easily discriminated message set.

Interpretation. If detected and identified, the next task for the user is to interpret the message. The designer wants the message to be either as intuitive as possible or to build upon cultural norms (e.g., word meanings, red is danger). It is useful to think of this process as being akin to translation. The device speaks to the user in its own language, such as tones and a blinking LED. The user needs to translate the message into his or her language, for example, "the battery is low." Although the user's guide or a quick reference card is designed to aid this process of translation, the designer needs to make the task of translation as easy as possible.

Three Qualities of a Successful Dialog

If the device's message is necessary, detected, identified, and interpreted the design may have succeeded in communicating, but to make the dialog more effective, *efficiency, association, transparency,* and *pleasantness* must also be designed in.

Efficiency. If the designer is convinced that a display is necessary, the next question to ask is *how* should the display communicate (e.g., quickly, accurately, or precisely) and *what* does it need to communicate (e.g., a value, the state of the device, a change of state, or a warning). The correct display often involves trade-offs. For example, if the purpose of a clock is to get the exact time, a digital display is better. If the purpose is to make relative judgments, an analog display is preferred. If one only needs to know when a certain time is reached, a blinking LED or tone may be best. The same analysis can be made with a display to indicate volume, FM tuning, or temperature.

The most basic functions should be indicated by the most simple displays. Is the device on or off? Is there a call on hold? Is there a warning? Two-state indicators (with an associated label, if possible) are recommended for these simple states. Two-state indicators are particularly useful as feedback to a user action, for example, pressing a key.

Often devices might be perceived as a bit too taciturn, particularly when a device is in an error state and the user needs more information. However, the opposite problem is just as prevalent. Designers of electronic devices will often present the user with much more information than they need, because it is possible to do so, or because a few users will need the information. E-mail message headers are a common case in point. The display should present only the information needed by the user—no more and no less. If the user does not need to know the exact temperature of the system, but only if it is high or low, then the display should be limited to this information. If the user only needs to know that there is a call on the line, but not its origin, then a simple ringer is sufficient. The role of the designer should always be to minimize the complexity of any dialog while maintaining the specified functionality.

Association. The device's output should be as closely associated with the user's input as possible. The classic case is the key click, which should occur less than 100 ms from a key press. Physical associations should be exploited as well. For example, a volume knob or slide switch that, by its very position, indicates the volume, is better than one that indicates its state via a display, especially if the display is physically removed from the input action.

Another form of association is to have the display itself serve as an icon for that which it represents. For example, the battery level can be indicated by a number, but a more meaningful representation to the user is a bar, perhaps in the shape of a battery, showing the relative remaining charge. Sometimes defining the association is easy. The LED should be lit when there is power to the device. Other times it is not so clear. Should an LED be lit when the microphone is on or muted? It might depend upon whether the activating switch is labeled "Mute" or "Mic." Human factors engineers determine these associations by iteratively designing and testing. The ideal display is one that is so intuitive that no training is necessary.

Transparency. Designers should not forget that the user's task is, for example, to place a call, record a message, or toast a muffin. It is not to operate a button, listen to a prompt, or "dialog with a device." If the user is aware of the device and its user interface, the interaction is likely to be cumbersome. The best user interface is the one that disappears. For example, on a traditional telephone, the handset is designed to look like something that can and should be grabbed in the middle and removed from its cradle. By simply picking it up, the user automatically opens the switchhook and requests service from the telephone company. The telephone company responds with a simple tone (dialtone), without a complicated prompt with various instructions. One can imagine any number of ways to make this interaction more complex, but as it is, it is a classic case of an invisible user interface.

Pleasantness/Attractiveness. Finally, the message (in its form, not necessarily its content) should be pleasant or attractive. Pleasantness is sometimes in direct opposition to alertability. If one wants the user to be aware that there may be a fire, one is not concerned with pleasantness as much as detection. However, in general, consumer items should be enjoyable and even fun to use. Displays should be designed with this in mind.

Visual Displays and Messages

Detection. As stated above, the visual message must first be detected to be useful. Detection is primarily a function of placement, brightness, contrast, and viewing angle.

Placement. Ideally, all displays should be positioned perpendicular to the user's line of sight. If this is not always possible, because, for example, the user might be sitting or standing, or the device might be on a desk or mounted on a wall, then the user should be easily able to tilt the display or the entire device. Displays are sometimes mounted in a recess in the housing for protection or for easier electrical connections. In these cases, the designer must be careful that information displayed on the edges of the display is not obscured by the bezel.

Because the display should not be obscured by the user's hands, keys the user may need to press while viewing the display should be placed below the display. A special placement issue arises with portable phones. Users cannot view the display and listen to the speaker at the same time (at least not with current designs). In this case, it is best not to display critical information on the display, but to use tones and voice output instead.

Brightness. *Brightness* is a subjective experience that is primarily a function of the *luminance* of an object and the *illumination* level of the environment. In a dark room (low illumination), a lower luminance level is needed to produce an adequate perception of brightness. It is not always the case that the brighter the display the better—for example, a car dash (at night) or a bedside radio. Displays must be bright enough to be read in the environment in which they are intended to be used. If the illumination levels cannot be controlled or predicated, as in a car, then luminance levels should be adjustable. The adjustment must be nonlinear, because smaller changes in luminance at low illumination levels will yield larger perceived brightness differences than the same luminance changes in high-illumination conditions. The standard recommendation is from $0.2\ \mathrm{cd} \cdot \mathrm{m}^2$ to $200\ \mathrm{cd} \cdot \mathrm{m}^{-2}$. (A sheet of white paper under standard office lighting has a luminance of approximately $160\ \mathrm{cd} \cdot \mathrm{m}^{-2}$.) When the environment may change in illumination, then ideally the luminance should be adjusted automatically, requiring a photometer of some sort. If the display is intended to be used in dim light, then purely reflected displays (LCDs in particular) must have auxiliary illumination. If the display is to be used in bright light, some technologies, for example, LEDs, may not be sufficiently bright.

Contrast. The higher the *contrast,* the easier it is to read the information displayed on the screen. Contrast is typically reported as a ratio of the brightness of the foreground over the background. A contrast ratio of 10/1 is excellent. A ratio of 3/1 is minimal. Color differences between the foreground and background contribute to what is called *color contrast,* and can also facilitate reading. In general, the closer two colors are in wavelength, the lower the color contrast. Such combinations should be avoided.

Resolution. On cathode-ray tubes (CRTs) and many displays, letters are composed of dots or *pixels.* The more dots per inch, the higher the *resolution* of the display. Higher-resolution displays can be read more quickly, and there is some evidence that lower-resolution displays cause more eye strain. The generally accepted rule of thumb is that, for English characters, a 5 by 5 matrix of dots is minimal, 10 by 10 is preferred, and performance peaks at about 15 by 15.

Viewing Angle. The *viewing angle* of a display is the angle off of perpendicular in which the display can reliably read. Liquid crystal displays can have very restrictive viewing angles (meaning they must be viewed "dead on"). Horizontal viewing angles have to be considered separately from vertical viewing angles. Consideration of viewing angle is particularly important when the user might approach a fixed device from a number of different angles.

Blinking. *Blinking* indicators should be used with caution. They can more successfully grab the user's attention than can color, shape, or the meaning of words themselves. This is good and bad. The blinking 12:00 on the VCR display is intended to warn the user that timed recordings will not be successful. However, it has become the number one example of bad human factors design, in part because the message cannot be avoided. In general, blinking means "attend to me now" and slower blink rates mean "less urgent" and fast blink rates mean "more urgent," with the blink rate of 0.5 s on and 0.5 s off being the approximate dividing rate between the two. Nonsymmetrical on/off rates, for example, 2.0/0.5 s, 0.1/1.0 s, and so on, can be associated with different states, but users have difficulty remembering the meaning associated with more than two or three different rates, undoubtedly because they are arbitrary in nature.

Discrimination. When a device displays a message to the user, the user must *discriminate* or identify the particular message from the set of all possible messages. The designer's job is to make this task of discrimination as easy as possible.

Reduced Message Set. When designing the range of messages a device can generate, it is often forgotten that each additional message makes *all* the messages in the set harder to discriminate. To take a common example, a simple charger may have a two-item message set: not charging and charging (LED off and on). A more complex system may have a three-state LED: not charging, charging, and fully charged (LED off, blink, and on). The three-state LED is undoubtedly still within the user's ability to discriminate. But what about a five-state system: not charging, slow charge, fast charge, fully charged, and problem with charging? Adding the two additional states may be warranted, but the designer should be aware that, unless done with care, the ease of reliably discriminating between the basic states, on and off, has been decreased. For cases in which precise and immediate discrimination is critical, for example, a warning or caution, minimal message sets should be used.

Color. *Color* is often used to increase the discriminability of the message set. To take the above example, charging states could be represented by a green LED, and problem states by a red LED. Although multidimensional discriminations are usually better than unidimensional ones, care should be taken that users with color perception difficulties may not be able to decode the message successfully. Color will be discussed more fully in the section on Interpretation.

Scales. The design of *scales*—the number of marks, the distance between them, their grouping, marker design, and so on—has been the topic of much research. As always, the goal is to present the information needed, no more and no less, in the most efficient manner. The design must begin with a human factors analysis, as described above. Take the example of a temperature gauge. Does the user need to set, read, or monitor the temperature? Is the exact temperature needed? Is the user likely to want to set it to a particular point? What universal symbology will the user likely be familiar with (e.g., blue as cool and red as hot)? The answers to these sorts of questions will drive the design.

Legibility. A special case of discrimination is the legibility of text on a screen. Contrast and brightness affect legibility, but in addition, the characters should be sufficiently large. Letters should be maximally distinctive. Using lower case letters, with true descenders (the part of a "p" and "g" that fall below an "o") and ascenders (the part of a "t" and "h" that rise above an "o") facilitates discrimination. The context in which letters appear facilitate identification as well. One reason using all upper case is almost always a bad idea is that this physical context (or word shape) of "WRITE" and

"RIGHT" is more similar than "write" and "right." Abbreviations are harder to read than full words, for several reasons, including this context effect. "LF" and "RT" have fewer cues to help the discrimination task than do "LEFT" and "RIGHT" (which is harder to read than "left" and "right"). A fundamental principle of information display is that the more discrimination cues given the user, the faster and more accurate the user will be. (See Osborne, 1995, pp. 96–105; and Sanders and McCormick, 1993, pp. 102–117.)

Text. To increase the discrimination of text, a few other recommendations from the literature are:

- Avoid reducing the spacing between lines to get more text on a screen. The minimal spacing should be one-thirtieth of the line length.
- Avoid setting text, all capitals, and abbreviations flush right.
- Minimize the variations with the text (font changes, bold, italics, color, etc.).
- Avoid too-small text. The height of the capital "H" should be one-two-hundredth of the average reading distance, with a minimum size of 2 mm to 3 mm. This recommendation should be adjusted for younger and older populations.

Interpretation. The function of a display is to support successful human–machine dialogs. Selecting the right display for the task and designing a maximally distinctive stimulus set is part of the job, but it should not be forgotten that the goal is not to be *noticed* and *identified* per se, but to communicate.

Color. Physical attributes of the visual signal, like *color* and blinking, can facilitate detection and within-set discrimination, but they are primarily used for coding the message. Users must learn these coding schemes if they are to interpret the message successfully. Users like color, and color can be very effective in both aiding in detection and in organizing and coding the device's message, with several major caveats.

- Color displays, in particular, LCDs, are more expensive.
- Cultural norms must be followed, for example, red typically is used for warnings. These cultural norms can vary throughout the world.
- 6% to 8% of the male population has some sort of color deficiency. For this reason, color should be used as a redundant coding of a message. For example, the common stop light uses both the color of the light as well as its position to indicate stop and go.
- The color of the indicator can change in different lighting condition. A dialog should never rely upon fine distinctions in color.
- Color can be overused, resulting in unnecessary visual complexity.
- The maximum number of colors that can be used in a coding scheme is about seven.

Symbolic Displays. Examples of *symbolic displays* are temperature and volume bars, which lengthen as temperature and volume rise. Symbolic displays can be very effective in quickly communicating low-resolution, nonnumeric information to the user. As always, cultural norms should be observed (e.g., empty, slow, low, old, or quiet is on the left or bottom of a figure), and designs should be tested, to make sure the user reads the display as the designer intended. There have also been a number of attempts to standardize *icons* for a number of operations and states, for example, play, record, stop, and pause on a tape player.

Pictorial Displays. *Pictorial displays* describe their referent visually and accurately. They are particularly useful when the message includes a reference to hardware (e.g., the location of a misfeed in a fax machine), there are well-known icons available (e.g., the representation of a bell to represent an alarm) or when the designer wants to avoid language biases. Once learned, pictorial displays can also help users locate a desired function more quickly than if they have to search through a field of messages. If pictures are to be used, they must be as true to life as possible, while removing all irrelevant information.

Multiple Cues. When the device needs to communicate to the user, more visual cues improves performance. Color, position, blinking, and content (if words) can all be used redundantly. The more critical the communication, the more useful is multidimensional coding.

Analog versus Digital. Much has been written about when to use analog (e.g., dials) and digital (number readouts) displays. The decision is application specific, but, in general, analog displays are preferred by users when changes in state must be monitored and when relative value is important. The ubiquitous low-cost, low-power character LCD display has encouraged designers to violate this rule, but newer, low-cost *graphical* LCDs are permitting more appropriate analog representations. Table 2 summarizes some of the reasons for selecting one type of representation over another.

Power Consumption

Power consumption may not seem like a human factors issue, but for devices powered by batteries, the human factors engineer must balance the need for a clear human–machine dialog and the need to have the device operate a long time without recharging. This often results in dimmer displays, timing out the illumination of a display, using a single indicator for multiple purposes, and perhaps even not using an indicator when an indicator would be warranted. This would seem to

Table 2. Tasks Best Suited to Either Analog and Digital Displays

Analog Display	Digital Displays
Estimation, relative value is important	Exact quantity, accuracy is critical
Perception of rapid change	Monitoring slow changes
Perception of direction change	Need to repeatedly adjust detail of scale
Controlling a value	Setting a value
Associated with analog input or control	Associated with digital input or control
Spatial information	Quantity, alphanumeric information
Quick access of trends	Quick access of value
Quick comparison of multiple displays	

be contrary to the desire to design an intuitive user interface, but it should be remembered that having to purchase a bigger, heavier, or more costly battery is also against the interest of users.

Auditory versus Visual Displays

When should the designer use an auditory display instead of a visual display in the human–machine dialog? Auditory displays are preferred for short, simple messages that will not be referred to later. They are also preferred when it is not certain that the user will be looking at the device. The classic case for an auditory display is the telephone ringer. Auditory displays should also be used when it is important that the user responds immediately or if the user's visual field is already overtaxed. A summary of these and other reasons are listed in Table 3.

Auditory Displays and Messages

The design of an auditory indicator is a surprisingly complex task. The indicator needs to alert the user, of course, but at the same time it must not be annoying to the user or to others nearby. The designer would like the meaning of the auditory indicator to be as intuitive as possible, but there are few cultural norms to follow. A pulsating, high-pitched tone may universally represent urgency, but what about a tone to indicate "please wait," or "your battery is low"? Fortunately, for many meanings (e.g., a busy telephone line), there are standards, although these standards are not always international ones. In consumer products, a balance must be struck between richer auditory feedback and more expensive speakers and drivers. As a case in point, it is known that pure tones are hard to localize, but sounds that contain multiple frequencies are more expensive to generate.

Detection. To be effective, auditory displays first need to be heard. The general principle designers follow to make stimuli detectable is to make them as different from other stimuli in the environment as possible. Humans are genetically engineered to rapidly perceive both uniqueness and changes in our environment.

Table 3. Tasks Best Suited to Visual and Auditory Displays

Visual Display Preferred	Auditory Display Preferred
Screen, but no speaker available	Speaker, but no visual display available
Speaker occupied by other primary activity	Display occupied by other primary activity
Noisy environments	Visual system is overloaded
Messages may disturb others	Private environment or only intended user will hear message
Quantitative information	Qualitative information
Information to be stored for later use	Warnings, information to be used immediately
User is actively looking at device	User may not be looking at device
User in close proximity to device	User may not be close to device
Message to be compared with another	Message independent of other information
Wide range of ambient noise	Wide range of lighting conditions
Long, complex messages	Short, simple messages
Nonlinear, menu-based interfaces	Linear, step-by-step interfaces
Scanning for relevant message necessary	Designer can predict what users need and when they need it

Loudness. It has been suggested that auditory displays must be at least 10 dB louder than the background noise to be reliably heard. There are a number of problems with this recommendation, however. Sound pressure decreases by the square of the distance; hearing loss is common among many users, especially older ones; and perceived volume is a function of frequency and duration as well as sound pressure (see Osborne, 1995, pp. 161–165; Sanders and McCormick, 1993, pp. 169–180). A smart device would behave like humans. People adjust how loud they speak based on their knowledge of the ambient noise level, distance from the listener, and perception of the hearer's abilities. Because consumer electronic devices are not as intelligent, the user should be able to adjust the volume of an auditory output.

At the same time, if the user accidentally has the speaker next to his or her ear, the tone should not cause hearing damage. General guidelines, given these caveats, are that sound levels should vary from 65 dB to 85 dB. How many volume levels should be offered and what should they be? For feedback tones, three levels are minimal—off, low, and high—and eight levels are the most that should be used to divide the distance between the very soft and the very loud. Each discrete setting must be distinguishable from its neighbors. Three dB is a recommended minimum.

Frequency. What *frequencies* should be used for maximum detection? Humans are most sensitive to sounds in the 300 Hz to 3000 Hz range. If the sound is to travel some distance or "bend" around corners, then lower frequencies are better. If there are other low-frequency sounds in the environment (e.g., speech) higher frequencies may be better. Alternating between two frequencies rapidly, like a telephone ringer, tends to stand out in most environments, as do tones that oscillate in amplitude (like sirens). Cheap frequency generators used in many consumer devices generate pure (single) frequencies. Such tones are notoriously hard to *localize.* The more complex the sound (and the lower the fundamental frequencies and the richer the overtones or harmonics), the easier the sound is to localize.

It should be noted that sounds can be intrusive. Not everyone likes them, and they can disturb others in the environment. In general, auditory displays should be made brief and pleasant, and the user should be able to turn them off.

Discrimination. The designer constructs a set of auditory displays to communicate meaning. Even the simple desk telephone has a set of two messages: ringing and nonringing. The more simple tones there are in the set, the harder it is to make them easily discriminated. (If one uses speech, the number of potential items in the set is very large.) Most people have difficulty discriminating, in isolation, more than two or three *frequencies, loudnesses, durations, intervals* (the difference in pitch between two tones), or *warbling rates* (the rapid alternating between two tones) within the ranges that it is reasonable to use these variables. Users are somewhat better with *melodies, rhythms,* and *timbre;* however, timbre (e.g., the difference between a violin and a flute playing the same note at the same volume) is harder to generate inexpensively and, unless the melody is very abstract, users become quickly annoyed by the repetition. For these reasons, it is best not to rely on the user having to discriminate among more than 5 to 10 different tones, regardless of how they are constructed.

Interpretation. Although there seem to be some universals of sound perception (e.g., loud sounds and high frequencies denote an emergency), nonlinguistic sound is nonrepresentative. It may seem quite intuitive to some users that a rising sequence of notes means, say, "open file," and a falling sequence means "close file," but other users will have the opposite intuition. The association between nonlinguistic tones and their meanings must be learned by experience. On the other hand, tones can be brief and nonintrusive. Verbal prompts (e.g., Your phone is ringing. Line two) can build upon the user's extensive linguistic knowledge, allowing large stimulus sets to be easily built, but such displays should be used with caution. They are more costly to implement, and based on the experience from both the automotive and consumer products industry, they are strongly disliked by many users, especially after repeated exposure.

Although some of the basic methodologies and design issues the human factors engineer considers when designing a human–machine dialog for a consumer product have been summarized, there is no recipe for a successful user interface. The particulars of the product will drive the details—particulars like prior product interfaces, competing products, development and product cost, development schedules, new technologies, the feature mix and feature interaction, applicable standards, interoperability requirements, product evolution plans, manufacturability, reparability, aesthetics, reliability, and changing market or customer expectations. With so many competing interests, the job of the human factors engineer is as much an arbitrator as a designer. Above all, the human factors engineer's job is to perform up-front analyses of the customer's needs, to be cognizant of the range of human factors issues that need to be addressed, to prototype, and to test repeatedly.

BIBLIOGRAPHY

P. Bullinger, M. Kern, and M. Braun, Controls, in G. Salvendy (ed.), *Handbook of Human Factors and Ergonomics,* New York: Wiley, 1997, pp. 697–728.

A. Chapanis, *Human Factors in Systems Engineering,* New York: Wiley, 1996.

J. Dul and B. A. Weerdmeester, *Ergonomics for Beginners: A Quick Reference Guide,* Washington, DC: Taylor & Francis, 1993.

D. Hoag, Computer input devices and the elderly user performing a fine graphics manipulation task, *Proc. HFES,* 1996.

W. MacDonald and A. C. Lowe, *Display Systems: Design and Applications,* New York: Wiley, 1977.

R. McMillan, R. G. Eggleston, and T. R. Anderson, Nonconventional Controls, in G. Salvendy (ed.), *Handbook of Human Factors and Ergonomics.* New York: Wiley, 1997, pp. 729–771.

T. Morgan et al., (eds.), *Human Engineering Guide to Equipment Design,* New York: McGraw-Hill, 1963.

D. J. Oborne, *Ergonomics at Work: Human Factors in Design and Development,* New York: Wiley, 1995.

S. Sanders and E. J. McCormick, *Human Factors in Engineering and Design,* New York: McGraw-Hill, 1993.

D. Wagner et al., *Human Factors Design Guide,* Atlantic City, NJ: US Department of Transportation, 1996.

JOEL S. ANGIOLILLO
Philips Consumer Communications

HARRY BLANCHARD
AT&T Labs

EDMOND W. ISRAELSKI
Ameritech

DESIGN, LOGIC. See ASYNCHRONOUS LOGIC DESIGN.

DESIGN METHODS FOR DISCRETE TIME SYSTEMS. See DISCRETE TIME SYSTEMS DESIGN METHODS.

DESIGN OF CONTROL SYSTEMS. See CONTROL SYSTEM DESIGN, CONTINUOUS-TIME.

DESIGN OF DATABASES. See DATABASE DESIGN.

DESIGN OF EXPERIMENTS. See RELIABILITY VIA DESIGNED EXPERIMENTS.

DESIGN OF EXPERIMENTS IN RELIABILITY. See RELIABILITY GROWTH CONCEPTS AND TESTING.