Conf-941001--2

SAND94-2548C

Top Ten List of User-Hostile Interface Design

The ten most frequent mistakes made in human-computer interface design

Dwight P. Miller Ph.D., CPE

Sandia National Laboratories
Albuquerque, New Mexico

Introduction

> ***Ergonomics** (or Human Factors Engineering) is a discipline concerned with designing machines, operations, and work environments so they match human capabilities and limitations (A. Chapanis, 1965)*

This report describes ten of the most frequent ergonomic problems found in human-computer interfaces (HCIs) associated with complex industrail machines. In contrast with being thought of as "user friendly,"[1] many of these machines are seen as exhibiting "user-hostile" attributes by the author. The historical lack of consistent application of ergonomic principles in the HCIs has led to a breed of very sophisticated, complex manufacturing equipment that few people can operate without extensive orientation, training, or experience. This design oversight has produced the need for extensive training programs and help documentation, unnecessary machine downtime, and reduced productivity resulting from operator stress and confusion.

Ergonomic considerations affect industrial machines in at least three important areas: 1) the physical package including CRT and keyboard, maintenance access areas, and dedicated hardware selection, layout, and labeling; 2) the software by which the user interacts with the computer that controls the equipment; and 3) the supporting documentation,

---

[1] I prefer to use terms such as "ergonomically sound" rather than "user friendly" because the latter is overused in the marketing of products that may or may not employ sound ergonomics.

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

administrative controls, and training associated with using the equipment. This paper focuses on issues involved with the second area: the software user interface.

In addition to pointing out the ten most frequent problems encountered, advice is offered on how to avoid the problems and create usable, efficient, and benign HCIs. This guidance comes from scientific research, HCI design guidelines, and the author's 20 years of experience in dealing with human-machine interaction. Make no mistake; this is by no means an exhaustive prescriptive tome on HCI design. That would take several hundred pages of text and figures, which no one would read. This paper is based on a report written for SEMATECH and the microelectronics industry by myself and my colleague and friend Hugh Whitehurst [ref.1].

An attempt was made to make this paper concise, readable, and useful. [Longer, more in-depth treatments of human-computer interaction can be found in the bibliography.] I feel that this small, but important, set of issues will help people remember them, and strive to use them in their own product designs. I hope that this document will raise awareness and stimulate the application of ergonomic principles to HCIs across your industry. I feel that if all of the problems identified in this paper are successfully avoided, and ergonomically sound HCI methods are used instead, the resulting HCI design will be better than 90 percent of the current supplier products available in industry.

Number 10. "Involve an ergonomist?..we don't know how to spell ergonomist"

The large majority of industrial HCIs never get the benefit of ergonomic considerations during the design process.In years past, lack of awareness was the principal reason. However, with the recent addition of the word "ergonomics" to the US culture's lexicon through product advertisements and generally increased awareness, inadequate application of ergonomics, for whatever reason, is usually the source of the problem. The following erroneous perceptions about ergonomics often prevent it from becoming a serious contribution to design:

1. Design requirements don't need to address the ergonomics of HCI design.
2. Ergonomists will check the design for flaws after some screens have been developed.
3. Ergonomics will negatively impact design schedule.
4. The design engineers can handle HCI design themselves.
5. The quality of the HCI will not affect the sales of the product.
6. Ergonomics is just common sense.

HCI issues are typically left to the engineers. Managers tell their staffs: "Make it smaller, more functional, and more reliable than last year's model in less development time......and, oh yeah, make it user friendly!" Since most of the more sophisticated industrial machines are controlled by computers, this task usually falls on the software engineering department.

Human Error

We have learned from experience with human/machine systems that the human operator, while being the most intelligent, can also be the least reliable component in the system. The two largest nuclear power accidents in recent history (Three Mile Island and Chernobyl) were found to be directly attributable to human error. Of all the aircraft accidents investigated by the National Transportation Safety Board in recent years, more than 70% were cited as attributable to "pilot error." This fact should not lead to an argument to remove the human from the system. The human's ability to adapt to change, recognize patterns, and make decisions with incomplete data suggest retaining human involvement, but assigning tasks at which they are adept.

Human Performance

Many factors affect human reliability and performance in the context of an industrial human/machine system: human interface design, training and documentation, work conditions and job stress, intelligence, motivation, perceptual abilities, etc. Except for the first two, all of these performance factors are either brought to the work situation by the worker, or are controlled by the employer, in the form of work hours, pay, job design, etc.

What remains, the human interface, the training, and the documentation are the only factors controlled by the equipment supplier.

Ergonomists apply what is known about human performance to the design of systems. This information has been empirically investigated in the field of human experimental psychology, and involves how people perceive environmental stimuli, cognitively process information, access memory, make decisions, and respond to external events.

Effective Ergonomics

There are four basic ingredients to the successful involvement of ergonomics in interface design: 1) motivation or buy-in on the part of the system designers, 2) solid expertise on the part of the ergonomist, 3) early and continuous involvement in the project, and 4) commitment to implementation of ergonomic recommendations. Equipment that employs sound ergonomic design can reduce training time and costs, reduce operators' errors and work-related stress, speed up recipe editing, improve human reliability, and hence overall system reliability and availability. Down time can also be reduced dramatically if the equipment is easy to calibrate, diagnose and repair. In addition to enhancing overall system performance, good ergonomics can also have a beneficial impact on marketability and sales.

Number 9. **"Read the manual?...na...we'll just call Dennis--he's good with computers"**

The lack of intuitive, consistent diaog with the machine causes prolonged searching through user manuals for answers, unnecessary frustration, increased human errors, longer learning curves, and generally less efficient operation. This section offers some basic strategies to make HCIs more intuitive and friendly.

Minimize Required Learning/Memory

The techniques outlined below all have the same psychological goal: to minimize the need for learning and the use of memory on the part of the user. The HCI developers should try to take advantage of any skills or knowledge the users bring to the interface. This is done by maintaining consistency, taking advantage of population stereotypes, using interface metaphors, providing structure, and being obvious.

Consistency

One of the primary reasons that people do not like to use computer systems is the lack of consistency and integration. This lack of consistency results in users having to remember different techniques to accomplish the same goal. The input required of the users and the output of the system should be consistent across the displays, operations to be performed, etc. The system should perform in a predictable manner without exception. The more internal consistency found in an interface, the fewer rules have to be learned by the user.

Population Stereotypes

Population stereotypes are expectations of relationships that develop from experience in a given culture. In the West for example, the following stereotypes exist for control/display relationships:

Concept	Direction of Display/Control
increase	up, right, clockwise, fill
decrease	down, left, counterclockwise, unfill
on	up, clockwise, lighted

off	down, counterclockwise, extinguished
forward	up, away from operator
backward	down, towards operator
left	left, counterclockwise
right	right, clockwise

HCI designers should take full advantage of this knowledge and be careful to avoid using interface tools that contradict these stereotypes.

Interface Metaphors

Many good interfaces succeed at being easy to learn by using an interface metaphor. A metaphor is a representational set of objects and procedures that takes advantage of a well-understood (and often overlearned) process or model to make a new, similar process more easily understood. Components of the new process or model are often represented by symbols of the old process so that the operations (actions) are comprehensible through analogy. For example, the Apple Macintosh® personal computer uses a desktop metaphor as its primary directory interface. Icons representing documents can be put into folder icons for storage or can be moved into a trash can icon when no longer needed. The desktop operations are meaningful and easily understood, based on experience with office materials, whereas the techniques of computer technology are foreign to most people, and would require additional training for comprehension.

Being Obvious

An article in *Business Week* [2] described the problems caused by high tech gadgets cluttered with unwanted features, "buttons from hell", and accompanied by complicated instructions. Consumers have become disenchanted with such products. After interviews with equipment designers, the authors wrote that, "Industrial designers and manufacturers have discovered that there is an inverse [relationship] between the reading needed to learn how to operate a product and the use of that product." There are a few design principles that, if followed, can make

---

[2] Nussbaum, B., and R. Neff. "I Can't Work This ?#!!@* Thing!" *Business Week*, April 29, 1991.

a product easy and enjoyable to use."And all the rules boil down to one thing: Be obvious. A machine should be designed so that customers can look at it, understand it, and figure out how to use it--quickly."

Number 8. **"This thing is so simple, a child of 8 could operate it"**

A Fire Department in New Mexico bought a $300,000 computer system that was too complex for many of the users (voluntary firefighters) to operate. The system, which was bought in 1989, was intended to link 11 fire stations and streamline inventory control, record keeping, and accounting, among other things. But in 1991, the computers in some of the fire stations were "unplugged and gathering dust."

Although a system was required that voluntary firefighters could easily learn and use, the system the Fire Department bought was, "complicated, making it hard to learn, and is inadequate for the operations it was intended to do." One Fire Chief complained that the system was getting a bad rap, that it wasn't really hard to use. For that one individual, and for the few he claimed to have trained, the system was acceptable. The level of complexity of the interfaces, however, should have been designed to meet the needs of the average voluntary firefighter with no computer experience.

An old adage of interface design suggests that the HCI should be designed to accommodate the least capable user (least common denominator approach). Alternatively, I recommend another approach that does not slow down the advanced users: stratify the HCI so that each successive level of sophistication of the user is matched by the interface style and tool set intended for that user. Some of the potential user populations for equipment containing vacuum systems include:

Engineers
Assembly operators
Laboratory technicians
Sales people
Maintenance technicians

Research scientists

Although they may overlap somewhat, the backgrounds and types of tasks performed by different user populations are very different and consequently require different sets of HCI tools. The tools sets should build upon one another so that the higher technical-level users (engineers, technicians and scientists) are familiar with the tools used at the lower technical levels (assembly and sales).

Number 7. **"I could tell you what the machine is doing if only I knew how to ask it"**

One of the problems found in many sophisticated machines is the omission of readily available machine status information in the HCI. The HCI should provide relevant machine status information with little or no input from the operator. A good test for this property is the "Walk up and see" test. If a potential machine user can walk up to the machine and gather relevant status information from the machine just by looking, then the HCI has an adequate status display. Machine status should be placed along the top of the CRT display, with the most important element in the left corner. This location takes advantage of the U. S. population stereotype of reading top-to-bottom, and left-to-right (see Number 9). The status information should be displayed constantly so that no operator action is required to access it. If a screen saver is used to prevent phosphor burn-in, primary status information could be included within the moving graphic (movement cycle should be long enough to read status information).

Ideally, the general machine status information (e.g., running, idle, etc.) on the CRT screen should be large enough to be legible at a distance of 3-5m so that users can see at a glance which machines are available for processing wafers. Alarm states should be attention-getting with the use of flashing yellow or red color fields. Annunciation of urgent faults with auditory displays is useful, although somewhat controversial. Sound is a good attention-getter, but intelligent algorithms must be employed to avoid operator annoyance and account for possible hearing impaired users.

The remainder of the status information can be displayed for normal reading distances (about .5 to 1.0m). Activity and comment logs should be scrolling fields, with items listed in reverse chronological order so that the most recent events are the most available. If space allows, time should be presented with both analog and digital displays. Figure 1 represents an example of a poor status display. The information is buried in a crowded header below a set of pull-down menu captions. It is difficult to read, incomplete, and the terminology used is not very understandable.

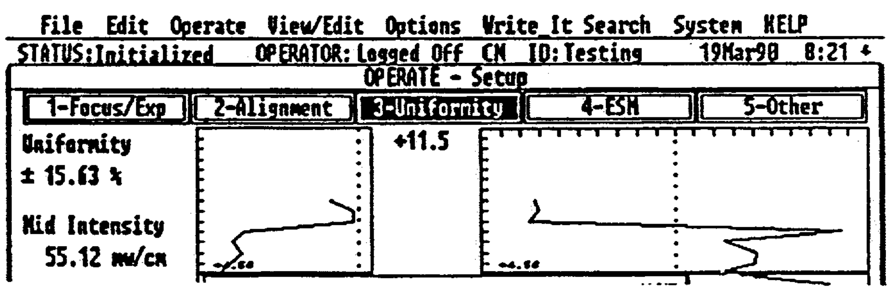

Figure 1. Example of a poor display of machine status.

In contrast, Figure 2 represents an example of a good status display. Its placement is in dedicated space at the top of the screen. It is formatted well, reading from left to right, uses graphics and colors appropriately, and allows for the accessing of additional lines of the activity log.

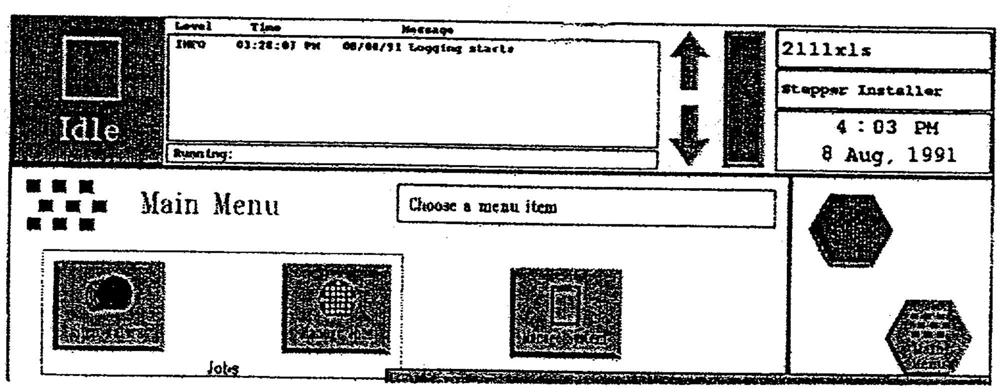

Figure 2. Example of a good display of machine status.

Number 6. **"Test it?...We didn't have enough time to design it!"**

No matter how experience or knowledgable, the interface designer (or ergonomist) cannot predict every aspect of an interface's level of acceptability by the user. Because each user population and individual user brings different expectations and skills to bear on the tasks, it is always desireable to iteratively design the interface through rapid prototyping and test for usability with a sample of the intended user population.

The first prototype should be a simple sketch of the general screen layouts. This information, if agreed to by the stakeholders of the design process, could form the basis of a design requirements specification document--a good idea for any HCI design project. An interactive prototype is best done on a easy-to-use software application that facilitates quick construction of screens and modifiable links between screens. Once a prototype is developed, future users can be asked to "operate" it and comment on their experiences. This process is used extensively in consumer-product design. Iterative cycles of design-test-redesign can help to develop a final design that has very high probability of satisfying most members of the user population. After being introduced, a similar iterative process can be used to take field experience and inject it into future improved versions of the design.

Number 5. **"What this thing needs, J.P., is some pull-down menus!"**

Many HCIs in industry are supported by, developed on, and run on computer workstations and sophisticated personal computers. They employ medium to large color CRTs, lightning quick microprocessors, and large amounts of active and archival memory. These machines were designed and developed to support computer-aided design (CAD), run simulations, perform statistical analyses, and do business calculations using spread sheets on PCs. Not one of these commercial products was designed to support a HCI for a machine on a manufacturing floor. Software packages and tools that are typically run on these machines employ "user-friendly" interface styles and tools including windows, icons, menus, pointing devices (hence the acronym WIMP), desktop metaphors, and color graphics. The engineers who design HCIs probably use many of these software packages in their work, and they should, because they are easy to use and produce great output.

Problems begin to occur when the HCI developers automatically adopt these [CAD/PC] interface tools in their own designs for industrial operations. They seem to be unaware that the manufacturing user population is not the same as professionally trained engineers, computer scientists, or software developers. The result is that HCIs for industrial equipment often are designed with these interface styles that are inappropriate for process control tasks (see Figure 3). This can lead to increased operator stress and inefficient machine control, and increased processing errors.

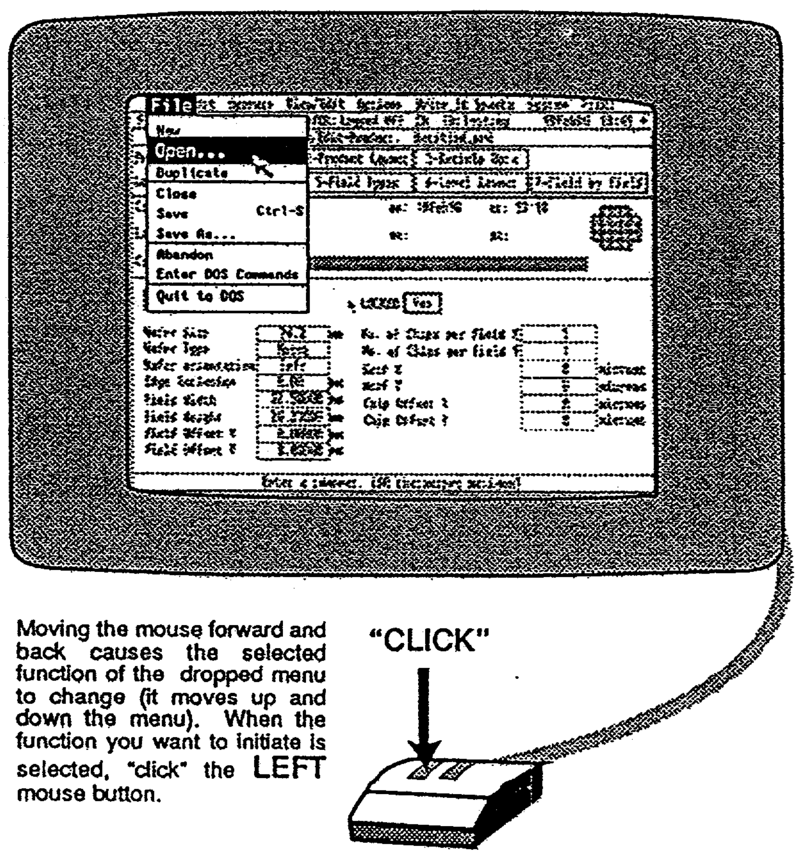

Figure 3. Typical CAD/PC style HCI with mouse and pull-down menus.

HCI developers should approach the design in its full context. The machine is in an industrial setting, used by operators of varied backgrounds and experience. The HCI should support the tasks of the operators. Crossman [Ref. 2] summarized the goals of [generic] process control operators into five areas:

1) to regulate or stabilize the process
2) to adjust the process in order to optimize it
3) to make changes from one product to another
4) to avoid breakdowns
5) to regain normal running after breakdowns

The following graphical HCI techniques are recommended over CAD HCI styles:

Control Panel Metaphors
Some of the simplest HCIs in process control present graphical representations of control panels on the CRT and use touch screens for input devices. This interface style is intuitive because most process control operators have seen and used hardware control panels with pushbuttons, backlighted legend switches, and LED indicators. This approach has been used extensively in "walk up and use" HCIs found in information kiosks, since the users presumably have no prior experience with the interface. A related technique is to allow for analog input of parameter settings via graphical slider controls (see Figure 4). An improvement is to provide a digital readout of the currently selected value next to the control.

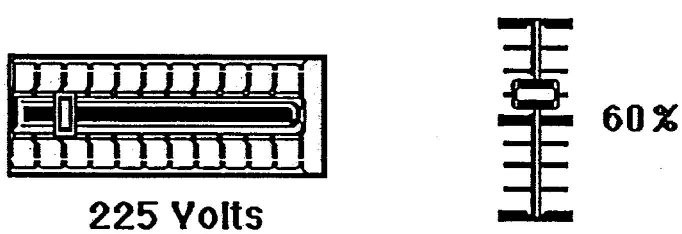

Figure 4. Examples of graphical slider controls.

I do not endorse the simplistic approach of displaying multiple, single-dimensional parameter values with graphical gauges and dials. That is usually a waste of valuable screen space. It would be preferable to use advanced display techniques such as the combination or integration of several parameters on one graph to facilitate extraction of the critical information. Integrated pitch and roll indicators in aircraft cockpits are better at giving the pilot situational awareness information than individual pitch and roll displays because the pilot does not have to do the mental integration of the two pieces of information. The addition of integrated graphical displays of system health has helped nuclear power plant operators overcome the user-hostility of control-room discrete displays for the same reason.

Mimic Diagrams

Mimic diagrams traditionally have been schematic diagrams of systems painted on control panels with all of the controls and displays located in places corresponding to the components they reflect or control. They have been very effective in process control environments because their isomorphism with entire system reinforces understanding of relationships among the system's components (see Figure 5).

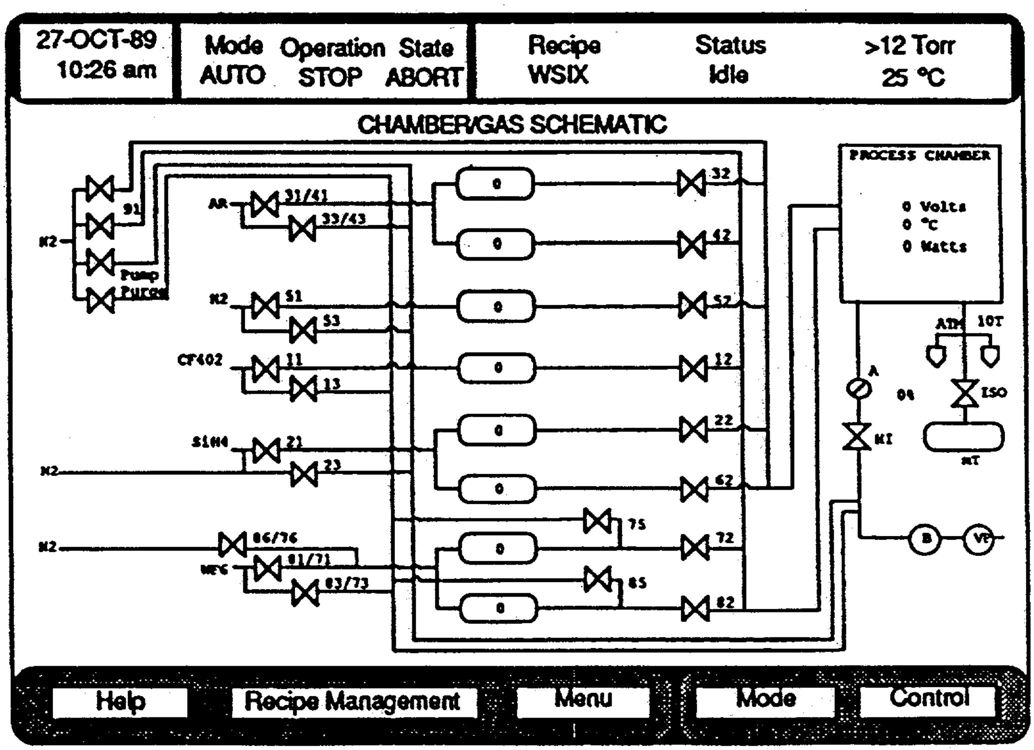

Figure 5. Example of a mimic diagram in a HCI.

Animated Process Mimics

Another method of displaying a system and its behavior is to show it graphically, in perspective, with moving parts and product. This type of display is especially useful if the operator cannot see inside the processing machine. The benefits are similar to the mimic diagram, but the major drawback is that only limited detail can be displayed for a complex machine. If parameter values are important, they can be collocated with the pictorial representations of the associated components, but at the expense of occluding part of the system diagram.

Number 4. **"The processor on this baby is so powerful we can produce 256 fonts with 16 million colors"**

Computing power is becoming more affordable and available with each passing day. Most of the HCIs in industry overuse the color capabilities of the computer workstations. Ironically, at the same time, there is extensive underuse of the vast graphical capabilities of these machines. These two problems may be symptoms of short lead times and lack of software developers in the design cycles. Well planned, pictorial interfaces take longer to produce than screens full of text and numeric data. This situation sells short the power of the HCI workstations, and short-changes the operators who have to wade through screens full of alphanumerics to get the information they need.

Use of Graphics

There are good reasons why people have so often said that a picture is worth more than a thousand words. Pictures are more compatible with the sensory, information processing, and memory abilities of humans than are words and numbers. Information is more easily extracted from meaningful pictures than from alphanumeric data; visual images can more closely match the real-world entities so less recoding of the information is required; and pictures are easier to remember than sentences and tables of numbers.

Icons

In computer based HCIs, graphical representations that symbolize the objects and actions they represent are called icons. Icons are metaphors for the entity that exists in the real world. Icons should include features that make their meanings obvious and look like what they represent. For example, icons used in the desktop metaphor of the Apple Macintosh® personal computer look like items found in a typical office, such as documents, folders, computer disks, and even a trash can. There is scientific evidence that people can more easily relate to icons than strings of text. Camacho, Steiner, and Berson [Ref. 3] investigated the effects on performance of alphanumerics compared to both monochrome and color icons used in pilot-vehicle interfaces. Icons produced faster reaction times compared to alphanumerics as the number of status displays increased. Subject errors were the same for alphanumerics and monochrome icons in one experiment, while alphanumerics produced fewer errors than color icons in a second experiment. This result may have been due to the precise meanings conveyed by alphanumerics. Overall, 23 of the 24 subjects preferred icons to alphanumerics. In another study, Steiner and Camacho [Ref. 4] used icons and alphanumerics that were displayed on a CRT to test the hypothesis that alphanumerics are more effective at low presentation rates and icons are more effective at high presentation rates. The authors found that when the displays contained only a small amount of information, there was no difference between icons and alphanumerics. As the quantity of information increased, subjects performed better using iconic displays.

Because people can relate faster to icons than text, but text has the advantage of specificity in meaning, it is recommended that icons and text be used in combination (see Figure 6).

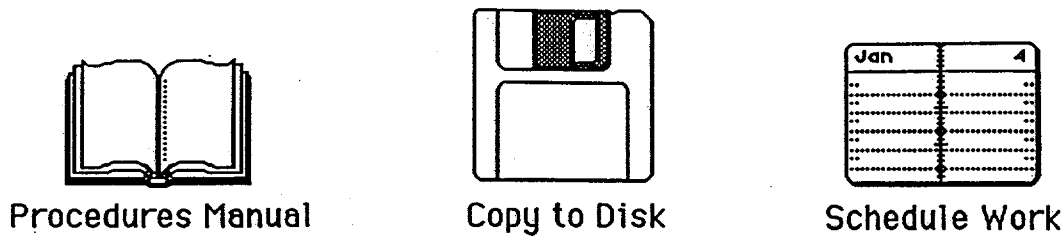

Figure 6. Examples of using icons with text.

Plots and Graphs
Graphical presentations, e.g., bar charts, histograms, and graphs, are pictures of data. They make it possible to present large quantities of complex data in a manner that simplifies comparisons, interpolations, and identification of trends (see Figure 7). Numeric data can be used to supplement graphical representations for precise check readings.

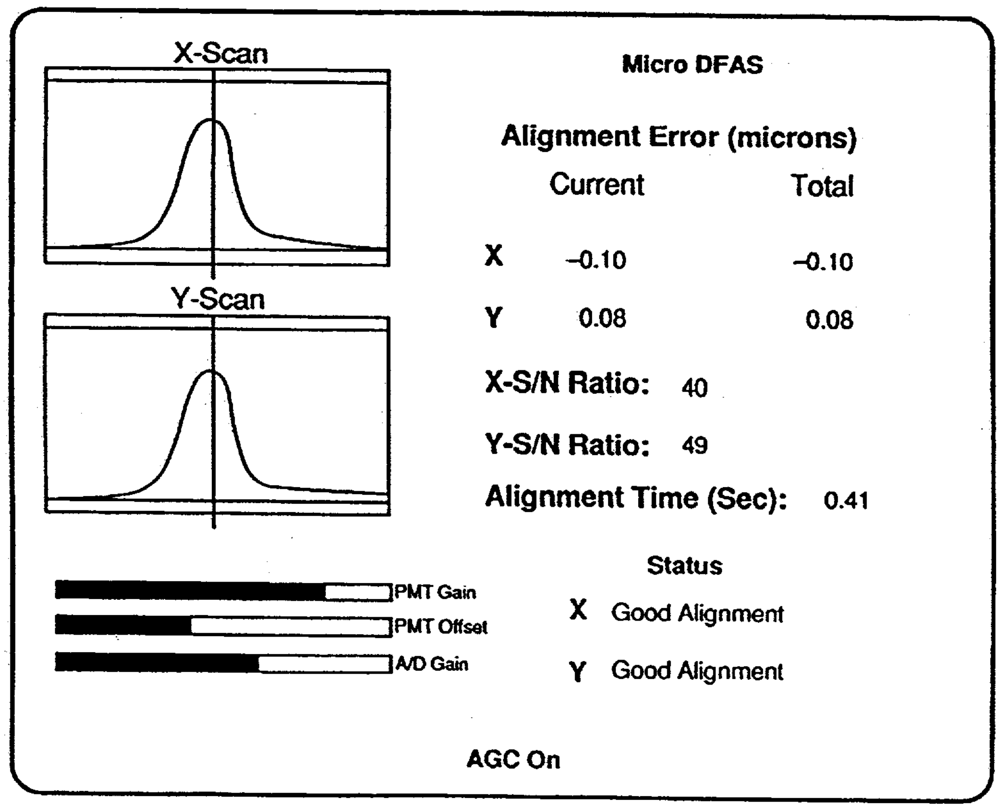

Figure 7. Example of a graphical presentation of data.

Use of Color

Today's high powered workstations and computers afford almost unlimited options for using color in the HCI. Although extensive color capability is great for color photograph rendition on CRTs, if it is used inappropriately in graphical HCIs, it can create busy, cluttered screens, irritate the user, and make information difficult to acquire. From recent observations at the trade exhibits, I feel that most equipment suppliers are guilty of overuse or inappropriate use of color in their HCIs.

If color is a display capability, use it conservatively in ways that can enhance HCI design and improve human-computer communication and interaction. Resist the temptation to create splashy, eye-catching screens that may attract a few unsophisticated customers at trade shows at the expense of irritating long-term users who have to deal with the HCI on a daily basis. Color is greatly overrated. Display dollars are better spent on screen size, grayscale, and resolution. Some very successful HCI designs eschew color completely. The original NeXT computer HCI used a high resolution monochrome display with many textures and levels of shading to produce a sophisticated look and feel that was very easy on the eyes. Keep in mind that roughly 8 percent of the male and 0.5 percent of the female population have color vision deficiencies that make it difficult to distinguish colors, most commonly red and green.

The best approach in designing a HCI is to do it in monochrome first, using all the textual, graphical, and pictorial techniques available in the software. After the design is completed in monochrome, if color is available, it should be considered a potential design enhancement tool to improve an already sound HCI.

Visual Discrimination Via Color

The first major use of color is to help the user visually search for, discriminate, and identify objects on a display. If hundreds of data points are displayed in a plot or graph, color can be an effective cue for finding a particular data point(s) of interest. Color can be used effectively to visually discriminate multiple lines on a graph or wiring and plumbing paths on a mimic display. Different functional areas of a HCI screen can be differentiated by different background colors, although graphical techniques such as boundary lines or brightness variations should be used whenever possible.

Color Coding

The second major use of color is to imply meaning. The most prevalent misuse of color in HCIs of IC-fab machines is the inappropriate implementation of color coding. Display designers often use color as a crutch when they run out of ideas on how to identify features of their displays. When done correctly, limited color coding can be very effective at providing meaning. The following guidelines should be followed when using color to imply meaning:

1. Color coding should be redundant with some other display feature, such as text or symbology.

2. When possible, color meanings should be consistent with traditional color expectancies or population stereotypes:

Color	Meaning
Red	Danger, inoperative, unsafe, hot, stop
Yellow	Caution, alert, hazard, abnormal state
Green	Normal, operating, safe, ready, go ahead
White, lt. blue	Neutral status indicators for system conditions

3. Color coding in the HCI should be consistent with that used in the environment, materials, and hardware associated with the user's job. Use real-world metaphors when possible, such as blue for water and brown for wood.

How to Design a Color Display

The trouble with most guidelines articles like this is that they describe elemental pieces of a large solution space to avoid or to use, but do not go further and define good regions of the solution space to help the designer develop a good display. Here is some advice:

1. Construct the basic screen layout using shades of blue and/or gray. Use a relatively light background (neutral gray or light blue), with darker objects and text (dark gray or black). This will help reduce specular glare from overhead lighting.

2. Following the principle of a "green board", keep information on nominal operations in the shorter wavelengths (blues and greens), and reserve the longer wavelengths (yellows, oranges, and reds) for abnormal-state information such as warnings and alarms. Use large patches of flashing "hot" colors to demand the operators' attention.

3. Use white text on a dark patch of background (inverse video) for unique or rarely-occurring, important information.

4. Keep the active information to only 25-50% of the screen to avoid a cluttered appearance.

This style of screen design will have an understated, sophisticated look, will be easy on the eyes most of the time, and will grab attention when the system gets into trouble.

Number 3. **"I don't know what I just did, but it seems to be working."**

Just as when communicating with another person, people need to know when they are communicating with a machine that their inputs, actions, or commands are being received and understood. For every action by the user there should be a noticeable reaction by the computer. Paraphrasing from Engel and Granda 1975, in Helander [Ref. 5]:

> Feedback to user action covers keeping the user informed of where he is, what he has done, and whether it was successful...immediate feedback by the system is important in

establishing the user's confidence and satisfaction with the system. One of the more frustrating aspects of any interactive system is sitting at the terminal after entering something and waiting for a response. Questions arise such as, 'Is the system still going?', 'Did the computer lose my input?', "Is the system in a loop?' A message that indicates that the system is still working on the problem or a signal that appears while the system is processing the user's input provides the user with assurance that everything is all right.

Feedback Timing

In general, the faster the computer response, the better. Variability of response times to certain transactions should be minimized for consistency and meeting the expectations of the users. The following list is a condensation of several guidelines found in the literature:

User Input	Maximum Response Time (secs)
Control activation, movement (keyboard, touch panel, light pen)	0.1
Pointing, sketching, appearance of a character or simple graphic on screen	0.2
Pull down menu, page/list scroll	0.5
Request for new/next page	1.0
Function command, simple inquiry, status request, list selection, identity check, commonly used error message	2.0
Complex inquiry, graphical update	5.0
Process execution, program/file load, complex graphical display	10.0

The information in Table 2 should not be misconstrued to suggest that an operator can wait 10 seconds for feedback on a complex request. The feedback should build upon itself--the input action (button selection) should elicit the most immediate feedback (shading change), the cursor should change to indicate processing, the command should then be processed and the screen should update to reflect it, or a message should be displayed informing the user of a waiting period.

Graphics Techniques are Best

Since most HCIs are visually oriented, feedback in response to operator input should take advantage of this orientation and use primarily graphics or pictorial feedback. Soft keys are software-generated graphic entities in a HCI that emulate control buttons. They can function as momentary pushbuttons, toggles, latches, selectors, or in any other mode. When selected, a soft key should immediately change graphically to acknowledge its selection. Subsequent activities depend on the type of switch it is emulating. If the soft key emulates a momentary pushbutton, it should revert to its original state shortly after being selected. If the soft key emulates a toggle, or a latching contact pushbutton, it should maintain its new graphic state after being selected, until it is selected again. Graphic transitions that indicate selection include those shown in Figure 8.

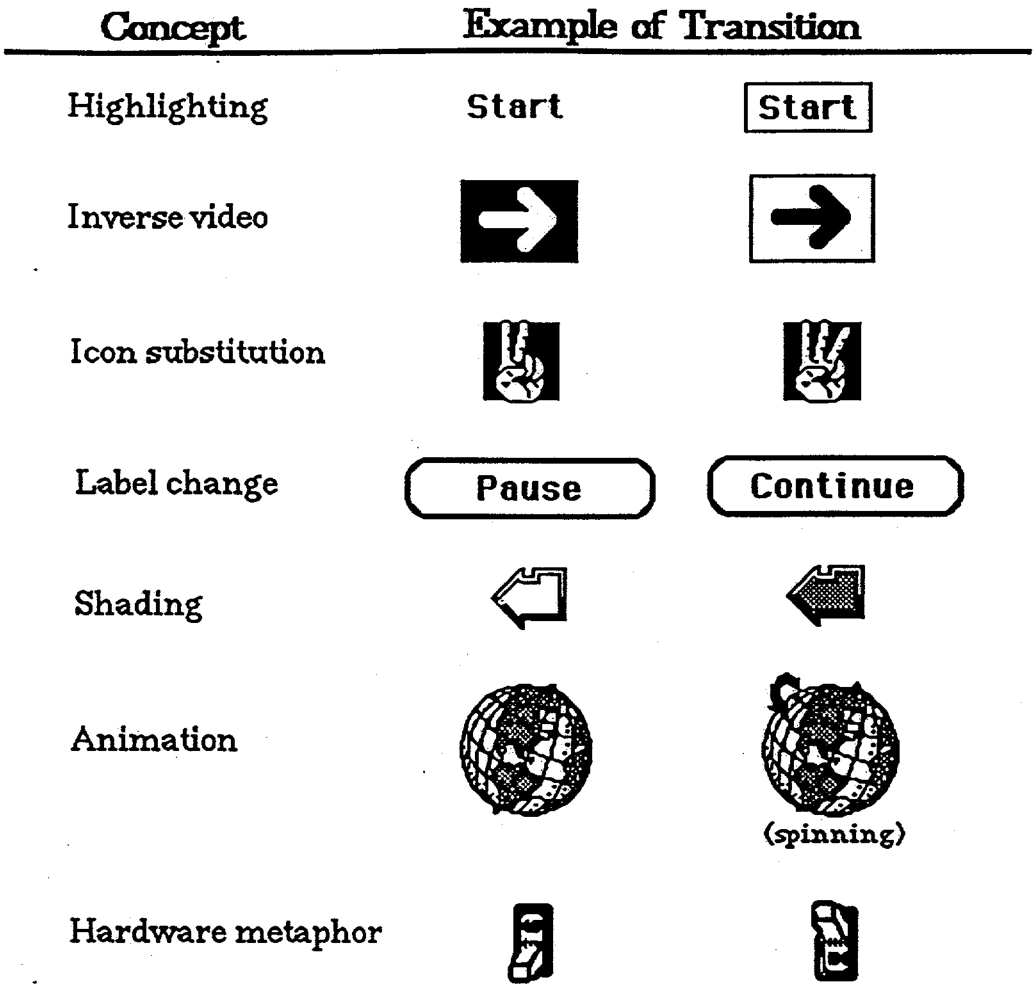

Figure 8. Examples of graphic feedback upon soft-key selection.

Tasks Requiring Substantial Time

When an operator asks a machine to perform a task that takes more than 10 seconds, the HCI should provide feedback on the milestones and the anticipated duration of the task. This information allows operators to budget their time accordingly and possibly leave the area to do something else. When a number of identifiable operations is not excessive, the HCI could list each milestone as it is completed. For example:

Load lock doors closed

:

Pump down in progress

:

Goal pressure achieved

:

Cleaning process has been started

For very long tasks, an estimated time of completion is helpful. Frequent updates of the display are required to indicate ongoing activity to the operator. Figure 9 shows an example of a percent completion display.

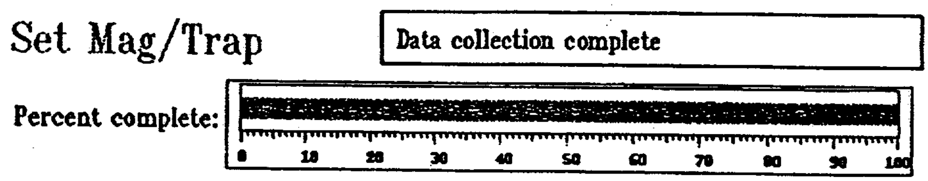

Figure 9. Example of milestones and percent task-completion feedback.

Error Feedback

Feedback is particularly important when an operator has made an input that is not expected, or cannot be interpreted by the system. The following guidelines apply to error messages:

1. Error messages should be worded as specifically as possible as to what is wrong and what can be done about it.

2. The wording should be consistent, in the user's language, and appropriate to a user's task and level of knowledge.

3. The messages should be polite, not obnoxious, without implications of blame to the user.

4. Users should be prompted to reenter only the portion of a data or command entry that is not correct.

5. Identification or reference numbers should accompany the text of an error message to facilitate looking up in a

reference document and keeping track of frequency of occurrence.

Number 2. **"One word: QWERTY!"**

The alphanumeric keyboard is an input device for computers that is a vestige of the desktop printing device known as the typewriter. It is an excellent tool for communicating strings of letters and numbers when they are known to the user. Early computers were designed to respond to either programs written via keyboards or commands typed in followed by a RETURN key (vestige of typewriter carriage return). The menu-selection approach, used in most modern HCIs, has the computer remember all of the possible relevant commands so the user simply points to the one desired. In spite of this HCI development, the keyboard remains the dominant input device for most computers and computer-driven machines that are human-operated. I think that most keyboards, especially the full alphanumeric keyboards (QWERTY, alphabetic, and Dvorak), are intimidating devices for novice users. Laboratory studies (e.g., Foley, Wallace, and Chan, 1984, Ref. 6) have demonstrated that keyboards require high levels of cognitive, perceptual, and motor processing, and lead to higher error rates than pointing devices.

Eliminating the Keyboard through Creative HCI Design

The operator tasks identified above as requiring keyboards for successful completion can be redesigned so that a keyboard, with its inherent propensity for errors, can be replaced by other input devices.

1. Personal identification can be done more quickly, with fewer errors, and with much greater security if the operators use their identification badge or some other personal identifier rather than typing in their access code. Current advanced techniques include: a signature on a digitizing tablet, retinal scan, hand-shape recognition, voice recognition, transmit/receive badge readers, magnetic strip readers, and bar code recognition.

2. Leaving log messages can be done more quickly and conveniently using digital audio recording devices.

3. Adjusting numerical values can be done effectively by providing pop-up slewing controls on the screen when the number field is selected with a pointing device (see Figure 10). Slewing up or down can be done by pressing the up or down arrows.

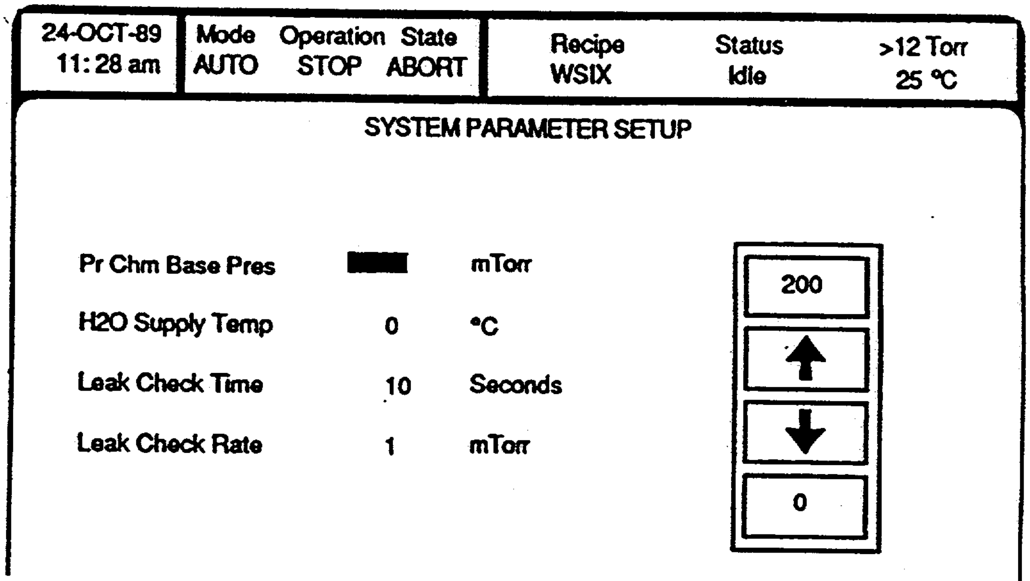

Figure 10. Example of pop-up slewing controls for adjusting numerical values.

Number 1. "Where the #@%! am I ?"

The sophistication of equipment and the current practice of using one or two CRTs to display all of the HCI contribute to one of two problems: either the screens get very crowded with information, or too many screens are used to represent the system. Often the former fault is avoided, due to guidelines that suggest limiting screen information. This leaves the solution of using many screens that often rely on complex structures relating them and defining their access paths. Large, complex structures are difficult to navigate through without adequate navigational tools or substantial experience. When developing a HCI of more than a few screens, provide the user with consistent, obvious, useful navigational tools that allow quick, easy access to any screen in the hierarchy. A simple, rather than a complex system structure is the starting point for quick understanding and ease of use. Simple structures, combined with good navigational tools can avoid most navigation problems.

Menu Selection

Most modern HCIs require navigation across screens by sequential selection of choices on menus. In large systems with many screens, only a portion of the available screens can be represented on any given menu. This necessitates multiple menu selections to navigate to many of the system's screens. In general, this 'depth' of menu structure should be avoided if possible (Miller, 1981, Ref. 7). When depth is unavoidable, the following techniques should be used to avoid navigational problems and help recover from wrong turns and wasting time on unwanted screens.

Providing Structure

People seek structure or organization in their environment, even in cases where no structure exists. (In those cases, structure is often created.) In computer-based environments, people seek to determine the structure of the system and build a mental model of the structure that they discover. Internal representation forms the basis of the user's understanding of the system and, in turn, his or her decisions and actions. The structure, and resulting mental model provide rules for situations previously not experienced by the user so that using inference, the operator can make the appropriate actions. The structure should be simple so that users can infer it from system interactions. Figure 11 is an example of a simple structure.

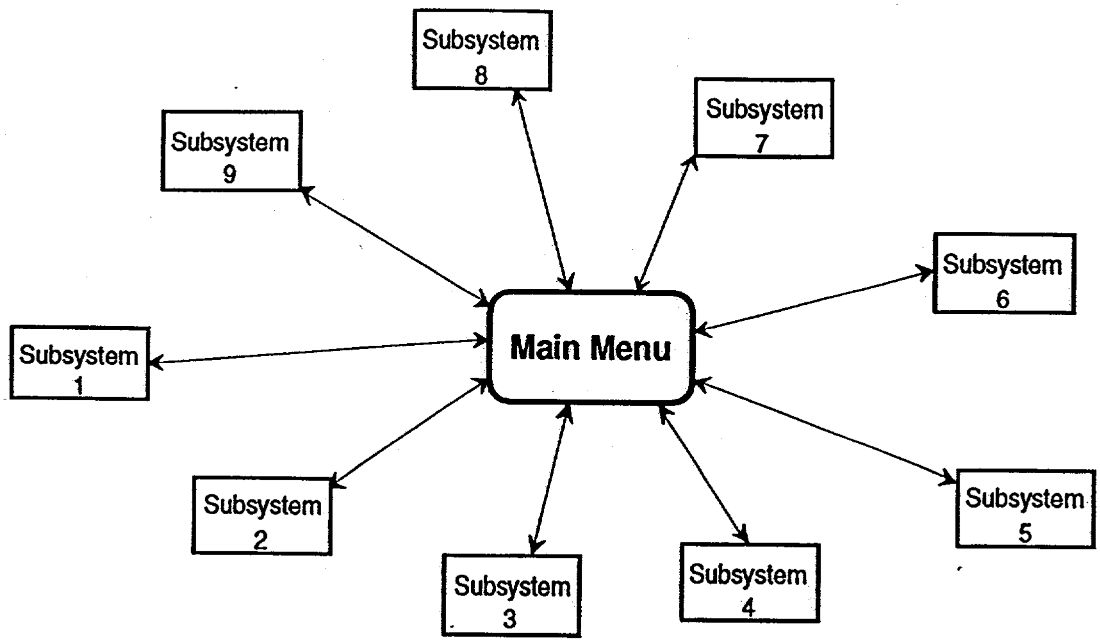

Figure 11. Relatively simple HCI structure

Maps

Drawings of HCI structures can aid HCI navigational performance. When used in orientation documentation, they give the user relational knowledge just as road maps do. They are best used in deep structures when ambiguity is high at high level nodes. Small graphical HCI structure maps displayed on the screens can be very helpful if the user's current location is included (see Figure 12).

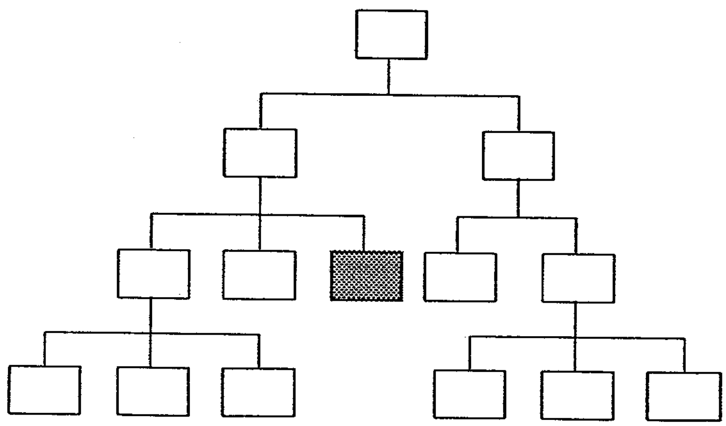

Figure 12. Example of a map display for a hierarchical screen HCI.

If a map is not possible due to space limitations, a simple history of previous screens, listed by name somewhere on the current screen, might suffice (see Figure 13). Some HCIs designers put the path information in the header of the current screen or window.

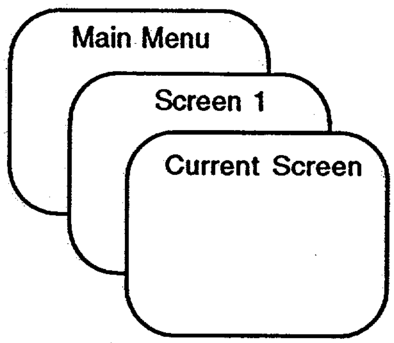

Figure 13. Example of a display of screen history or path to current screen.

Retreat Keys
The easiest way to undo a recognized screen navigation error is to cancel the erroneous (previous) input. Thus, an **undo** or **backstep** key can be very useful. Figure 14. illustrates two versions of backstep keys.

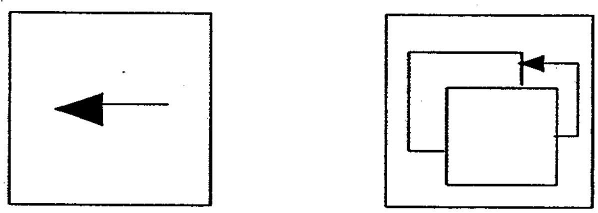

Figure 14. Examples of iconic backstep soft keys.

Main Menu Keys
We have found that a '**main menu**' or '**go to top**' key is extremely convenient for hopping from the current location to the top of the screen hierarchy. This type of key is useful in all systems, regardless of size or complexity. Figure 15 shows one version of a main menu key.

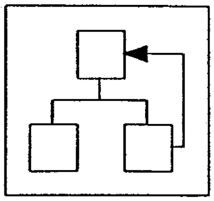

Figure 15. Example of an iconic main menu soft key.

Direct-Access Keys

Several functions or screen locations may be considered important enough or used frequently enough that they deserve their own direct-access keys. Just as critical functions such as ABORT or STOP have dedicated controls, or soft keys, critical screens such as ALARM LOG or HELP may need to be accessed quickly and easily. Direct access keys should appear in the same location on all the screens in the HCI.

Epilogue

My goal in accepting the invitation to address the professionals within the American Vacuum Society was to raise awareness and stimulate the application of ergonomic principles to HCIs across your industry. For a more in-depth treatment of the issues raised, please consult Ref. 1.and some of the better texts on this topic listed in the bibliography, and most importantly, consult with an ergonomist or a human factors engineer (you can call the Human Factors and Ergonomics Society in Santa Monica for information on consultants). Good luck in your future HCI designs!

References and Bibliography

1. Miller, D. P. and H. O. Whitehurst, *Preventing User-Hostile Interfaces in IC-Fab Equipment: Ergonomic approaches for preventing ten frequent user-interface problems*, SEMATECH Technology Transfer Report 92091299A-ENG, November 1992.

2. Crossman, E.R.F.W. "Automation and skill." in E. Edwards and F.P. Lees (eds.) *The Human Operator in Process Control*, Taylor and Francis Ltd., London, 1974.

3. Camacho, M. J., Steiner, B. A., and Berson, B. L. "Icons vs. alphanumerics in pilot-vehicle interfaces," in *Proceedings of the Human Factors Society 34th Annual Meeting*, 1990, 11-15.

4. Steiner, B. A. and Camacho, M. J. "Situation awareness: icons vs. alphanumerics," in *Proceedings of the Human Factors Society 33rd Annual Meeting*, 1989, 28-32.

5. Helander, M. (ed.) *Handbook of Human-Computer Interaction.* Amsterdam: Elsevier Science Publishers, 1988.

6. Foley, J. D., Wallace, V. L., and Chan, P. "The Human Factors of Computer Graphics Interaction Techniques." *IEEE Computer Graphics and Applications, Part-1,* November 1984, 13-48.

7. Miller, D.P., The depth/breadth trade-off in hierarchical computer menus. In *Proceedings of the Human Factors Society 25th Annual Meeting*. Human Factors Society, Santa Monica, October 1981.

8. Brown, C.M. *Human-Computer Interface Guidelines.* Norwood: Ablex Publishing, 1988.

9. Galitz, W.O. *Handbook of Screen Format Design*-second edition. Wellesley Hills: QED Information Sciences, 1985.

10. Gilmore, W.E., Gertman, D. I., and Blackman, H. S. *User-Computer Interface in Process Control: A Human Factors Engineering Handbook.* San Diego: Academic Press, 1989.

11. Salvendy, G. *Handbook of Human Factors.* Chaps. 12.1-12.7, Selected Applications of Human Factors in Computer Systems. New York: John Wiley & Sons, 1987.

12. Smith, S.L. and Mosier, J. N. *Design Guidelines for User-System Interface Software.* Tech. Rept. MTR-9420, The Mitre Corp., Bedford Mass., Sept. 1984.

13. Thorell, L.G. and Smith, W. J. *Using Computer Color Effectively.* Englewood Cliffs: Prentice Hall, 1990.