User Interfaces for Casual Users

Honours Project Report
by

Craig D Farrow

1993

Contents

Chapter 1

Introduction

The interaction between people and computers is an important field of study, and is a high profile topic in computer science at present, with much still to be learned. The user interface is the medium through which users and software communicate. From it the users get their initial (and often lasting) impressions about the software, and it is this part of the program that either frustrates the users, or helps them to make full use of the underlying functionality.

The main thrust of this project is in the area of user interfaces for situations where users have little or no time to learn lots of detail about how the software should be used, but need to be able to do what they want with a minimum amount of learning. This class of user is referred to in this report as a *casual* or *novice* user. There are many principles for developing better interfaces, and a lot of these can be applied, some in an extreme way, to the design of interfaces for such casual use. A lot of these ideas may seem rather obvious, but if these issues aren't consciously thought about, and the principles applied in the design process, then it is very easy to produce a product that is more difficult to use than is necessary.

The main situations where computer interfaces must cater for the casual user are where the general public has access to such things as public information kiosks or teaching exhibits. Here people will often use the system only once, and will want to be able to achieve or learn something useful in a very short time. A few people may use it more often, but what they want to achieve will be very basic.

There are many examples of products that aren't explicitly computer applications, but that still involve an interfaces by which users control its functions. These products are mostly intended for use by the complete novice, and yet they are often too complex for people, and so only the most basic functions are used. Examples of such devices are photo-copiers, video recorders, stereo systems, microwave ovens, and clock radios.

Even for more complex applications it is desirable that the system should be at least *usable* by a casual user, even if the full functionality is only accessible to those who have more experience or have taken the time to learn more. Ideally users should be able to quickly pick up the basic operation of the product, because most people don't want to spend hours learning how to use a device or program, but want to be able to get their job done as fast as possible. Also those who are infrequent users will want to find it easy to remember, with little or no re-learning, how to use the product when they return to it. Examples include re-setting the time on a digital clock after a power-cut, or doing a quick calculation on a calculator or spreadsheet.

As a more specific area of interest, this project looks at science displays for a science centre where people, primarily children, interact with the exhibits to learn about various scientific principles, and to have fun at the same time. Visitors have paid to be entertained and want their children's interest in science to be stimulated, and so it is important that exhibits are created that are appealing, interesting, instructive, and very easy to use.

The first part of this report presents various ideas and principles in human-computer interaction (HCI) and relates them to the casual use situation. Much research and literature on the topic deals with HCI in general, particularly software designed for more long-term use, although some material relates directly to the novice user.

As part of the study of interfaces for casual users a few science exhibits were observed in a science centre environment. Their use is discussed in Chapter 3 with comments on their suitability and effectiveness.

Chapter 4 discusses the details of a small science display that was developed to investigate the design process, particularly when designing for the novice user. Various alternative interfaces were considered, and Chapter 5 reports on some brief evaluation experiments with them.

Chapter 2

Human-Computer Interfaces for Casual Users

The study of Human-Computer Interaction (HCI) is concerned with promoting principles for better design of computer-based products, making them more *usable* so that users can be more effective in accomplishing their tasks.

Usability can be hard to define, although a few people have proposed some suggestion. Norman [20] comes from a psychologist's point of view and says

> *A device is easy to use when there is visibility to the set of possible actions, where the controls and displays exploit natural mappings.*

Here he is saying that in order for something to be easy to use the actions that can be made must be *obvious* to the user so that they can easily discover and remember how to use the device or system. This can be illustrated with a door that has a large handle on it for gripping. The nature of the handle suggests that the door should be pulled, and if this is the case then people can just go through the door without thinking. It is easy to use. However, if the door can't be pulled (as the handle suggests should happen) then there is an unnaturalness that misleads people into pulling the door first. People find such a door to be a nuisance because naturally we want to do what is intuitive and obvious, which in this case was the incorrect action.

Preece [22] suggests that an interface has good usability if it can be used safely, effectively, efficiently and enjoyably. This last point is an important one because if the users find an application boring and hard to use then their motivation for work will diminish along with productivity.

For large and small applications alike it should be the designers' goal to make them intuitive and easy to use. If more complex programs and devices, as well as those intended for strictly casual use, are designed for the novice user then the result will be something that is much more usable for everyone.

The sections in this chapter address the three fundamental aspects of HCI — the human, the computer, and the interaction between the two — as well as some additional topics that do not fit into any one specific category. These topics are discussed in relation to use by casual users, particularly in a science centre environment.

2.1 The Human

Learning about the potential users and the tasks they need to accomplish, and then designing for them right from the conception of the design is one of the most important aspects in the development of any product [23] [22] [27] [14]. Right from the start the abilities, experience, personalities, and needs of the users must be taken into consideration so that the majority of the user population is accommodated. Experimentation and consultation with potential users is necessary to ensure that the system is designed from *their* viewpoint. What may be obvious to the designer can often be far from obvious for the typical user [23].

The vast diversity usually present amongst potential users for a science exhibit makes this difficult. In essence the lowest level of ability and experience must be assumed and the system designed with this in mind.

The following aspects are important and need to be considered in the design.

Motivation

Different users have different reasons for using a system, and this depends both on their personality and the type of program being used. A science exhibit needs to attract and keep the interest of users while other types of display or utility are simply there as a tool to help *already* interested people complete a task. Examples of the latter are doing the washing, and finding information about a specific geographic area. In these situations users want to be able to carry out operations quickly and efficiently, but on the other hand don't want a system that is restrictive to use because of its simplicity or lack of flexibility. A balance often needs to be made between the amount of functionality provided and the simplicity of the interface, although if both can be combined then this is even better.

For a science exhibit the users will generally have little time or patience to work out how to use a complex interface, and their initial motivation is often low when they don't even know what the exhibit is about. Therefore the initial impression is vital, and users must be able to achieve something interesting very quickly and easily.

Ability and Experience

The major difficulty in designing for the user is the large amount of variation in ability and experience of users, both in computer use and in the specific area of application. This presents difficulties in design because people at both extremes of knowledge and experience must be made to feel comfortable using the system.

Ideally a system shouldn't assume any special knowledge of computer systems, but some applications may require the user to have some basic knowledge of the subject matter. For very specialised applications the user community will often be fairly narrow in its characteristics and area of expertise and this makes design simpler because a specific *type* of person can be targeted and the system can do its job better [21]. However for systems that are intended for novice users the job is particularly difficult because of their timidity and inexperience. Experiments have shown that first time users can take up to 100 times longer to perform tasks than frequent users [27].

Some systems, such as a public information kiosk or a library catalogue, need to be easy to use for the novice, but also need to provide additional power for those who are more experienced, or who gain proficiency through regular use. Doing this allows more frequent users to make more efficient use of their time. There may, however, be some conflict here with wanting to provide power to the frequent user *and* ease of use for the novice or infrequent user. This needs to be achieved in a way that doesn't confuse those who only use it once or occasionally.

One way of allowing for diversity in the user population is to provide different levels of abstraction, with either the user or the system itself choosing an appropriate level to operate at. Doing this allows more frequent users to make more efficient use of their time, while helping casual users to cope with the complexity of the system. Novice users generally have to be carefully led through tasks by the system to ensure that their operations go smoothly, whereas more advanced users will want to take the initiative and "do their own thing". There may be some conflict here with wanting to provide power to the frequent user *and* ease of use for the novice or infrequent user. In terms of a science exhibit, the complete novice must be catered for and there is little need to provide more power. This simplifies the design process because only one end of the continuum needs to be considered.

Preferences

Again there is great diversity in what different people prefer. This may be in the control methods used, or simply the look of the interface. Often there may be cultural differences where a person's culture and upbringing may influence their tastes or ways of doing things. For example, Americans will generally perceive a switch to be *off* when it is down, whereas people from some other cultures will tend to think of down as meaning *on*. Problems arising from these sort of issues can be avoided with neutral alternatives such as using switches that move sideways or in-and-out, rather than up-and-down.

Other cultural or personal differences are apparent in the association people make with certain colours, imagery, sound or words. Sensitivity is needed in the interface design, especially where people from many different backgrounds are to use the system.

In a reasonably sophisticated application differences can be catered for by making the system easily customisable and extensible to give the user control over the environment they use [7]. For the casual use situation, however, there is little or no scope for user customisation so extra care and testing is needed to produce an interface that conveys the correct meaning and can be understood by its users.

2.2 The Computer

The second component of HCI is the computer. Of course there doesn't have to be a fully fledged computer involved. Many consumer electronic goods such as VCRs and micro-waves are microprocessor based, but its presence is heavily disguised behind the buttons and LCD displays that form the interface. Novice users often have anxiety about using computers [27] so achieving this *transparency* of the computer will make users more comfortable thus enabling them to focus more clearly on what they are doing. This is a key factor in usability in that we want the interface to "almost disappear" [30] so that users can concentrate on their work, exploration, or pleasure, rather than the individual actions needed to carry out the task. In other words, the interface should not get in the way of the job to be done [21].

The following sections on presentation, consistency and standardisation relate to interface design issues regardless of the purpose of the system or the form of the interface — a computer screen or a panel of physical buttons.

2.2.1 Presentation

The presentation of a system is through its user interface. It is here that the look and feel of the tool is established, either to make it a delight to use or to make it a nuisance and frustration. The interface must present information and possible actions in a clear and understandable manner while showing enough detail of what is happening. A reasonable amount of effort is needed, then, to ensure that the system is well presented, always keeping in mind the user's perspective.

Cushman and Rosenberg [9] give lots of very specific guidelines for various aspects of the presentation including sizes of labels, and the use of colour and sound, as well as recommendations for input and output methods. These guidelines are useful for ensuring that sensible choices are made for even the most basic of interface elements. A well designed interface in all other aspects can be made a frustration to use just because, say, the text used is uncomfortably small.

The following sections cover some of the more specific issues involved in the interface presentation. They relate to all types of product, but science exhibits should have particular emphasis on attracting and keeping attention. The focus is on interfaces presented on a computer screen, however, the general principles apply equally to other forms of display.

Colour

The use of colour is a difficult issue that applies to both screens and other aspects of an interface as well. Generally, careful use of colour can greatly enhance the presentation of a system by making it more appealing to users (especially children), showing extra detail, and by highlighting important areas. Colour can also help acceptance of a system, but often it is not necessary or performance improving [15]. Generally colour should be used for subtle details or for highlighting something of interest. In some circumstances the colours used may not be easily distinguishable because of lighting conditions or the user's colour-blindness. Therefore colour alone should not be relied upon to show information. Instead there should be redundancy so that if the users can't discern colour differences then they can use shape, size, position, or some other detail to identify objects. Shneiderman [27] suggests designing for monochrome first, getting the contrasts and groupings right, and then introducing colour. This will ensure that the system is still usable even if colour can't be made available or if any of the users are colour-blind.

Salomon [24] points out that colour has generally been under-used despite the available technology because it is so difficult to apply. Specific difficulties relate to the environment and external conditions that interact with the chosen colours, and affect perception. These are seldom static, making design even more difficult. Also the physiological and cultural diversity of the user population should influence the choices of colour.

Any choice of colour should be tested to see that it has the intended meaning because often there will be common expectations of meaning (such as green meaning "go") by users [27], and going against these expectations is likely to lead to problems in user understanding and interpretation. Where appropriate the user should have control over the colours that are used.

Graphics

The use of graphics in an interface gives much more scope than text and can provide greater visibility to functions than if purely textual input and output is used. In some situations more detail can be provided with graphics, and connotations often associated with words can be avoided. However, graphical interfaces haven't yet provided the complete answer for the user interface, with text still being a major, and necessary, part of the communication process. Text provides greater descriptive power and versatility, and will probably never be completely replaced, although graphics can be used to enhance the display and to present data in more appropriate forms such as graphs and pictures. Graphical output can also be used to provide more meaningful help or instruction by *showing* how something should be done [18].

In rushing onto the graphical user interface (GUI) band-waggon there is the potential for a graphical display to become too complex, so care is needed to keep the layout simple and uncluttered. Tufte [32] poses interesting thoughts on minimising non-data 'ink' and superfluous 'chart-junk' for paper-based data presentation. The general idea of reducing redundant clutter and distraction is good, and can be applied to computer screens or related displays as well.

Animation

Animation can be used in the interface to make images more informative and useful by using movement to illustrate the functionality or purpose of interface elements [2]. As with graphics, the use of animation places increased demands on resources and so the value of its use needs to be carefully evaluated. For more regularly used systems users may find animations too much of a distraction or hindrance to program performance, and so it may need to be possible for the user to disable it.

For science displays animation will add interest and may be an effective means of showing the user what they should do, what they can achieve, or what particular actions will accomplish.

Sound

Sound is another medium that can be used at the interface to provide enhancements to feedback, information, or instruction that is given. Audio output may involve simple tones, synthesised speech, digitised sounds, or combinations of these, all of which can enhance the atmosphere created by the interface. For a serious application the scope for sound is more limited, although there is potential for greater use of sound, particularly in the form of speech, for describing actions or providing more meaningful feedback.

For a science display, exciting noises can make it more appealing, and the use of appropriate sounds can set the mood for certain parts of the interaction. Movie makers and game designers have used sound to good effect for a long time, and their example could be followed to produce much more interesting displays.

The use of sound can often be irritating or embarrassing to the user, especially when it is used to attract attention to errors. In most circumstances errors made are not particularly critical and other less annoying means can be used to inform the user. In most situations sound should be mainly used to help draw the user's attention to something, or for embellishment, but not for conveying information that cannot be obtained elsewhere. This is because some people may find the noise annoying and turn it off (this should be possible where users may use the system for some time), and it is likely that some users will be deaf or partially deaf, thus hindering them in their use of the system if the design insists that they rely on sound for some things.

2.2.2 Consistency

The issues of consistency and standardisation are amongst the most difficult to deal with in HCI. At one extreme it is highly desirable, if not essential, that consistency is maintained within and between systems. Yet standards can be restrictive, with old technology or ideas persisting long after they have been outdated by better methods. The next section discusses examples where standardisation has been very necessary, and also where it has restricted advances and improvements in usability and productivity.

Consistency *within* a user interface is important, giving the user stability in their use of the system. If there are inconsistencies within a system then users have to exert more effort to remember what part of the system uses each method of interaction, and their actions are likely to be confused. Novice users especially will be put off by inconsistencies. Shneiderman [27] points out that consistency between user interfaces of different systems is also important for minimising learning time and errors for people who are already used to one particular user interface metaphor. Targeting such familiarity is useful for making it easier for people with some experience to learn how to use a new program. Once users have spent time learning a particular methodology for doing something, the majority will resist learning a new way of doing a similar thing because it means more effort on their part, and the original learning seems to be wasted.

The other side of the coin is that being too consistent can be constraining, with improved techniques not being used for the sake of being consistent. In the long run the user community will lose out from this course of (in)action. Despite the strong need to retain familiarity among interfaces, Tognazzini [31] points out that in the rapidly changing world of computers it is often hard to be consistent. Therefore the system designer must choose the most important aspects of consistency to maintain and improve on other areas.

Tognazzini offers four principles to consider when making modifications to an existing system or metaphor.

1. Don't change something unless it really needs changing.

2. If you must make a change then make it a large and obvious one. Visibility of any changes helps avoid confusion.

3. Add new skills to the user's skill set rather than expecting them to change existing skills.

4. Provide consistent interpretation of the user's behaviour. This is more important than consistent system objects or behaviour because users can cope with a different look, but it is unfair to change the meaning of their input, such as changing "D" to mean Delete instead of Draw!

These principles are directed at updated versions of a product, but they can also be applied when designing a new system with interface ideas and methods taken from other systems. Some may object to this, saying that this is breaching copyright, and yet it is pointless "re-inventing the wheel" for each new product. The example in the next section about car pedals illustrates how sometimes there just has to be cooperation and free sharing of conventions or methodologies for the user's sake.

2.2.3 Standardisation

Standards in product design may be put in place by companies or organisations to ensure that their products are consistent, or they may grow from popularity of a particular design that other designers see as good or popular with users. In the second situation it is often an early design that gets established as the de-facto standard, and it becomes very hard for designers to gain user acceptance for new and better designs. On the other hand, if standards are slow to develop then there may be so many different ways of doing things that global standardisation becomes impossible. [20]

The biggest barrier to changing or introducing standards is the investment of time and money that has often gone into existing methods. Most users who have already spent a lot of effort learning to cope with a system that may be deficient, illogical or awkward, are unwilling to change to something 'better'. People tend to be conservative, resisting change and improvement for the sake of stability and familiarity. Also with anything that is reasonably well established there will have been a lot of financial investment involved, and to even suggest that some things now be standardised seems to be foolish.

However the situation is not always so hopeless. Sometimes new ideas or technology find favour with the general public and old ideas can be superseded, or at least diminished in use, by them. An example is the digital clock. The advantages and novelty of the digital display seem to have been sufficient for it to take over from analogue versions in many situations.

The following "standards" are examples of how standardisation can provide consistency and stability for design and also how it can restrict the acceptance of newer and better ideas.

- A classic example of old technology having a strong foothold is that of the "QWERTY" keyboard. It was designed to restrict typing speed so that the manual typewriters wouldn't get jammed [9]. Now, with most keyboards being electronic, this is not a problem and yet the QWERTY layout has become the most prevalent one. The most notable alternative is the DVORAC layout that was designed to minimise finger movement and to put more work onto the right hand. Figure 2.1

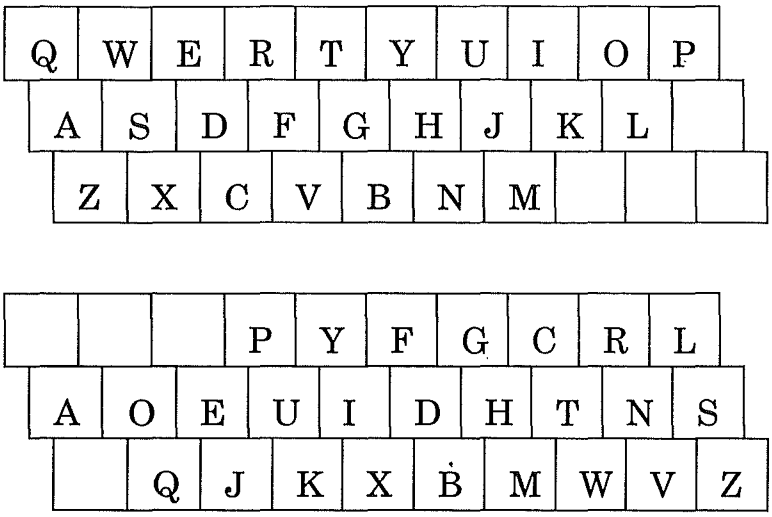

Figure 2.1: The QWERTY and Dvorac Keyboard Layouts

illustrates these two layouts (letters only). Note how the Dvorac keyboard has the most frequently used letters on the middle row, and that the right hand has the most work to do — the QWERTY keyboard design has the *left* hand doing about 57% of the work! This is an example of where a much better solution exists but, because of wide use of the older design, the better one isn't used very much. Anyone wanting to try the DVORAC keyboard has to spend some time in re-learning as well as having to overcome the problems of trying to use this layout on systems that are based around the QWERTY layout.

Some conventions in keyboard shortcuts lose their significance on the new layout. For example the `cut`, `copy`, and `paste` editing trio often use the X, C and V keys because of their proximity on the keyboard, yet using these keys on a DVORAC keyboard would not have the same significance.

- A good example of where standardisation is necessary is in the arrangement of the clutch, brake, and accelerator pedals in motor vehicles. After a reasonable amount of experience in driving, the basic operations of changing gears, braking and accelerating become fairly automatic. If each car manufacturer decided on their own arrangement there would be chaos when people tried to drive an unfamiliar vehicle. In this situation a pedal layout that works well has been developed, and there would be no purpose or advantage in changing to any other arrangement.

  A related case of different standards being developed is that of which side of the road people drive on. Some countries use the left side while others use the right. Although this causes problems as people travel to different countries around the world, the cost, inconvenience and confusion of converting a whole country to the opposite convention would seem to be prohibitive[1].

In designing science exhibits or other systems for casual users only, there is the opportunity to break away from restrictive standards or conventions to use improved methodologies. This can cause confusion though, if the difference is subtle, or if users are too familiar with another way of doing things.

2.3 The Interaction

The third component of HCI is the interaction itself. In order for any meaningful achievement there needs to be communication between the user and the computer. The user must be able to express their wishes about a task that they want to accomplish, and the machine needs to give suitable feedback to the user providing information, its current status and its requirements for more input. In the situation where the user has very little time to learn, or only uses the system briefly, it is important that the communication is logical and intuitive so that little effort is needed on the user's part to understand and remember how to use the product. Even for more experienced and frequent users ease of use is an important goal, although the requirements of the system may preclude making the interface as simple as desired.

Crawford [8] suggests that good games are an excellent example of interaction that works. Games are an absolute test of their interface because they are a leisure activity rather than a necessity, so people will only use them if they are enjoyable. If they are too slow, or too difficult to operate, then there are always other games to try. In contrast, applications such as word processors are used because they are faster and more convenient than typewriters. In the face of difficulties and a bad interface users may still persevere and not complain because they are still more productive with it, and feel that this is enough.

For a user to interact effectively with the system they need suitable input and output devices, adequate feed-back, appropriate metaphors through which they understand the system components, a good understanding of the model of operation, and helpful error reports. These areas are discussed in the following sections.

2.3.1 Interaction Devices

In order for communication to take place at the interface there must be some physical means for the human and the computer to instruct and respond to the other. Recent developments in technology are bringing more variety to the way humans and computers can communicate, however, it will take time for these to be fully explored and understood. Buxton [6] comments that no *one* device is suitable for all needs, so each application needs to be looked at individually, and the suitability of alternative devices assessed in relation to the tasks that must be performed.

[1] Apparently some countries have, or are considering, making this change!

Output

The primary method of output from computers has been the Cathode Ray Tube (CRT) that can display text and graphics, often in colour, has great flexibility in what is displayed, and can usually display a lot of information. There are other means of interactive output as well, such as LCD displays, lights, and sound. The use of sound can include digitised noises and speech. For the casual user the use of a computer screen may be off-putting so other less threatening methods could be used when there is a small amount of information to be communicated.

Input

Input to a system should be *natural*, and easy to use. Dumais [10] describes "natural" as something that naïve and casual users spontaneously do. If input devices operate in a way that is not natural then users will be more likely to make mistakes, especially when they are inexperienced or under pressure.

The QWERTY keyboard has been the most common means of input for a long time, providing easy entry of text. Joy-sticks and mice have also been used, particularly in games where simple input is often adequate. For the casual use situation, and especially for children, alternative and novel methods are needed because the keyboard, and even the mouse, can place too high a cognitive load on the inexperienced user. Even with other methods, only those "keys" that are necessary should be available, otherwise people may get over-whelmed [3]. Large buttons and objects that children can get their hands on should be used where possible. There is an excellent exhibit using very large buttons that act as a musical keyboard that people can stand on to play the notes. Here the designer has creatively made something fun to play on with the users being unaware of the "buttons" or the interface *per se* — they just play the piano — so transparency has been achieved.

When it comes to alphanumeric keyboards for casual use, the designers often use alphabetical layouts and keys with little or no movement. However it has been shown that for casual users different keyboard arrangements make no significant difference [20]. So whenever the full alphabet is needed a QWERTY keyboard should be used because this makes no difference to the inexperienced user, but it will speed up those who are even mildly proficient with a QWERTY layout. If the keyboard is intended for prolonged use then tactile feedback is essential — zero travel keyboards are only suitable for infrequent use [9].

Touch-screens can be a useful means of input for an on-screen based interface, and Shneiderman [27] recommends their use for situations where the users are novices. Touch-screens overcome the problems with mouse use by people who haven't had much experience, and particularly small children who find it especially difficult [13]. The use of touch-screens provides flexibility for the programmer, while also being durable and easy to use. However, prolonged use can cause muscular fatigue, especially if the screen is vertical [9], and the lack of tactile feedback tends to encourage *pushing* rather than touching, especially if the system is slow to respond. Other problems are that the user's hand obscures part of the display, and the screen gets covered in finger marks.

2.3.2 Feedback

Timely and clear communication from the system is vital so that the user is aware at all times of what is happening, and why. This includes making the current state of the system obvious, and the effects of each action visible [20]. Sometimes a particular input has no effect in the system and so it is appropriate for there to be no feedback. See the next section for more details regarding this *inaction*.

Leung [16] found that system response time is a major factor in usability, with users getting frustrated if their expectations of system performance were wrong. Casual users, particularly, will tend to be more impatient, and will be more adversely affected by inadequate or unsuitable feedback.

In a lot of circumstances the system's response will take some time and so the user's input will be ignored or buffered for a while. Whenever this is the case some indication needs to be given to show how long the process will take [9] [1]. This may be in the form of progress or percentage indicators that give the user an idea of how near the task is to completion. This information is comforting to the user, giving them reassurance that progress is being made, and it often makes the time seem to pass more quickly. For very resource intensive applications it may be necessary to accept longer delays, but the user must be made aware of this.

2.3.3 Error Handling

An error occurs when the user inputs something that the system doesn't recognise, or that has incorrect structure. Errors can occur as the result of the user accidentally doing something they didn't intend, or they may be the result of an incorrect understanding or interpretation of the system and what can or can't be done. Good system design through the use of appropriate models and metaphors, as well as suitable instruction, can help achieve the aim of minimising the potential for error [17].

Error notification usually take the form of *error messages*. These come in all sorts of varieties and are often cryptic and unfriendly, making the task of recovery very difficult or even impossible for the majority of users. If the designer has been thoughtful and considerate then they may be helpful and supportive, allowing the user to quickly continue with their work. Experiments have shown that specific and friendly messages improve the performance and attitudes of the users [26].

Lewis and Norman [17] [20] believe that eliminating errors completely is impossible and so the system should be designed to handle errors appropriately. This involves giving meaningful and useful information to the user when something goes wrong, and allowing them to remedy the problem if necessary. They also suggest that messages should be user centred, making the user feel comfortable, and the system should take the blame for any action that is not understood. If the computer is to take the blame, then we must blame the designer, for it is the designer who determines how the program behaves. So perhaps, if this view is taken, the designer may make more effort to reduce or eliminate the potential for errors instead of blaming the user. Through careful and creative design many types of errors can be avoided, and it may be possible, especially in simpler systems, to produce a design that prevents the user from making *any* invalid input.

At any time there will be a set of actions that make sense in the situation, and anything else that the user does can be ignored, thus avoiding many errors that the system

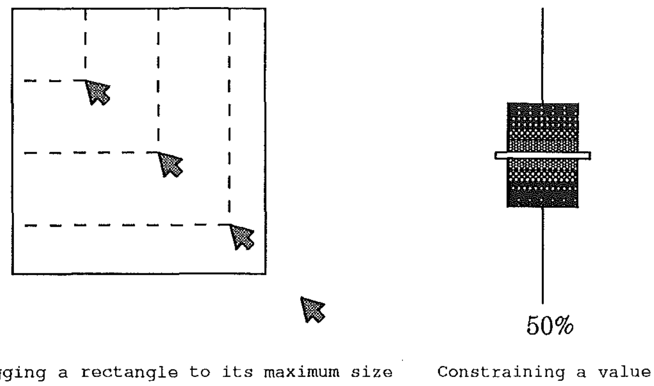

Figure 2.2: Preventing errors by constraining the user's input

may otherwise need to report. The set of actions available can be shown through the use of a *menu*. Menus provide visibility to the legal actions and reduce the mental load on the user [10] by helping them to remember what actions are possible [22]. Any user input in the form of keystrokes or mouse clicks can be ignored if it is not selecting one of the menu items, otherwise the requested action can be carried out [17].

The direct manipulation paradigm provides good scope for limiting the user's actions in natural ways. The first diagram in figure 2.2 shows how the size of a rectangle can be restricted by simply stopping at the maximum dimensions. This form of constraint doesn't have to be apparent until the user tries to go beyond the limits, in which case they restriction is passively enforced. The second diagram depicts a *slider*, which again provides natural constraints to an input value. The current value is shown below the slider, which can be dragged up and down to alter the value. This is much better than asking the user to type in a value and then having to validate it and report an error if it was out of range.

Finally, it is often helpful to the user to provide an *undo* function so that they can back-track if they want to. This also encourages experimentation because the user knows they can easily recover if something doesn't work satisfactorily. Whenever an operation cannot be undone the user should be warned beforehand, and given the option of cancelling it.

2.3.4 Models

Any interactive system or device can be thought in terms of a *finite state machine* model that has one or more *states* of operation, with *transitions* between states that are usually made as a result of user input. These states may represent *modes* of the system where its behaviour is different, such as "insert" or "overstrike" modes in a text editor. They may also be an intermediate state in a sequence of operations, or they can represent the value of some data in the program, such as the status of a toggle switch. It is interesting to note that it is possible to break the system down into states to an almost arbitrary level. Mostly, though, it is sufficient to only consider the uppermost one or two levels.

In many systems the structure can be viewed as a hierarchy of state machines. Some states may have independent state machines within them, each having its own *current* state. The global state of the system is the combination of each "current" state within another current state. Transitions can be made between higher level states independently of their internal state machines.

The reason for looking at the state model of a system is that the user often needs to know the current status of the system so that they know what the effect of their actions will be. If the user understands the underlying model, and the current state is made visible, then they will have a greater chance of using the system effectively. However, if the user's mental model of the system differs too greatly from the actual model, then they will tend to make errors, and draw wrong conclusions, as they attempt to match their view of the way the system works with reality. For casual use situations it is particularly important that the users have a clear understanding of how the system works. Keeping the model as simple as possible will make it easier to describe, and for the user to comprehend.

The term *model* can also be used to refer to a set of concepts and behaviour relating to a specific object or style of interaction.

2.3.5 Metaphors

> *Metaphors function as natural models, allowing us to take our knowledge of familiar, concrete objects and experiences, and use it to give structure to more abstract concepts.* T Erikson [12]

People often use metaphors to describe abstract or complex things in a way that is easier to understand. For example, we talk about the abstract concept of arguing with military terms such as "win", "lose", "attack" and "stand your ground". Similarly metaphors can be used to help users understand a computer system by providing translations from real-world concepts to the computer's concepts, abilities, functions and models.

In interacting with a device or computer system the user needs to be able to see what actions can be taken, and have an understanding of what the effect of each action will be. The use of appropriate metaphors can help the user to think about the operations in a way that they can comprehend, and to understand the underlying model of the system, thus making it easier to use.

When using a metaphor it is important that the mappings from the real world to the computer system are natural and meaningful. Bad metaphors decrease usability and cause difficulty if they suggest an incorrect model. As well as using a good metaphor to describe the model (one that doesn't lead to false conclusions about the operation), it is necessary to ensure that these ideas are clearly understood by the user, otherwise they will most likely come up with an inappropriate model that will lead to errors, frustration and decreased usability.

Often the designer can take advantage of physical analogies and cultural standards to come up with suitable metaphors [20]. These include real-world events, objects or institutions that embody some of the characteristics of the system that users find difficult to understand [12]. The use of graphical objects that represent real world objects is great for reducing learning costs and increasing user acceptance [1]. For novice users it is particularly important to use familiar concepts to help them to understand how the system works.

One of the most wide-spread graphical metaphors that has been developed is the "desktop" metaphor, which is used on many modern day systems in some form. Some of the concepts involved in this metaphor allow users to think of directory structures as nested "folders" or "drawers", letting them use concrete ideas from their world knowledge to help them understand a new, and often foreign, computer concept.

Part of the general desktop model is the concept of *direct manipulation*, which refers to the visibility of objects in the system and the way that they can be manipulated directly, usually with a mouse. There are, however, many deficiencies with this metaphor that can cause difficulty if it is misunderstood. Buxton [5] comments that direct manipulation is a big improvement on the old ways, but there is still a need for new ideas and metaphors.

Along with the desktop metaphor there are other associated concepts and metaphors that are often used. One of these is the idea of a "window" that users can "look" through into directories and applications. The use of windows as a means of managing multiple input dialogues generally works well, despite the fact that very few physical desk-tops have windows on them!

Another metaphor often used is that of a "trash-can", which is used for discarding unwanted files, directories, or other objects. The idea is that putting things into the trash-can doesn't discard them completely — they can still be retrieved from it if the user realises they shouldn't have thrown the data away. This seems to work well most of the time, however the metaphor can break down when people make false conclusions about the way it works. A feature of physical trash-cans that is usually omitted in computer systems is that of it being automatically emptied on a regular basis. Those who are familiar with putting rubbish into a real rubbish bin in an office, and then finding it empty in the morning, can get caught out with a trash-can that never gets emptied.

Perhaps the functionality of the trash-can could be extended to include automatic deleting of data contained in it that is more than, say, a week old. However, such a subtle change will cause confusion, and may be disastrous for people who are used to retrieving data from their trash-can that has been there for more than a week. Another solution that may help to solve this problem is to change the picture of the trash-can to indicate that it is not empty. This adds a visual reminder for the user that there is still something in it that must be deleted.

An interesting use of the trash-can on at least one system is to eject disks by putting disk icons into it. Surely this action will be misleading to users by implying that the user is throwing the disk away and doesn't want the data on it any more!

Figure 2.3 shows an atypical direct manipulation interface that highlights a few common components from systems that use this metaphor.

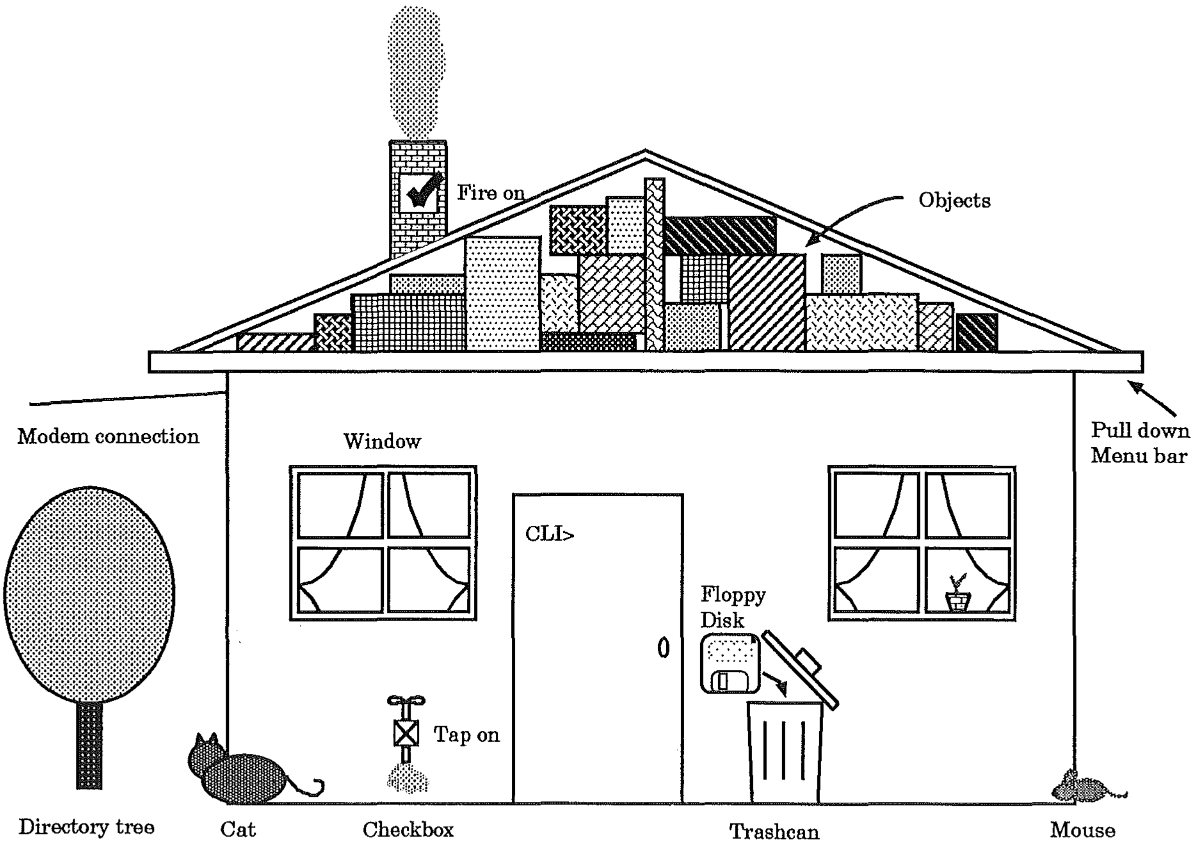

Figure 2.3: Components of a Desktop Environment

2.4 General

2.4.1 Iterative Design

> *There comes a time when one must stop suggesting and evaluating new solutions, and get on with the job of analysing and finally implementing one pretty good solution.* Unknown

An important aspect to interface design is the need for iteration [27] [14]. The first version produced, even with careful design and thought about the principles described in this chapter, will inevitably fall short of the best solution in various ways. Knowing this, it is important to allow time and money for the design–redesign process, and to consider how the program can be written in a way that is easily modified, improved and updated. Of course this cannot go on forever, so at some point a final version should be decided upon.

Part of this iterative process must involve significant user input and testing to discover deficiencies and flaws in the designs. This makes the whole development more costly and slower, but each iteration is a refinement (or complete re-design!) based on the previous one, so the final result must be better than if this testing and iteration was neglected [27].

By paying proper attention to HCI principles and doing rigorous testing there will often be reduced costs and faster development [30], although some redesign will usually be inevitable in order to develop a worthwhile product.

2.4.2 Testing

Testing the interface design must be an integral part of the whole software development cycle. At all steps, every aspect of the interface needs to be tested with "real" users — the designer knows too much about the product and so is useless for testing [23].

Poor interface design can be costly, in terms of time, money or even lives [22], but often systems are badly designed, requiring excessive effort by the user just to operate the system! Extensive testing and evaluation needs to be carried out to ensure that the user's needs are taken into account, and that the interface is natural, enabling the user to easily understand and use it.

Virzi [33] found from experiments that about 80% of usability problems were found with four or five test subjects, with the most severe problems likely to have been detected. This means that in the initial stages of development only a few test subjects will be needed to find the main problems. Then towards the end more thorough testing can be done to iron out the less significant problems.

2.4.3 Evaluation

Evaluating the effectiveness of an interface can be quite difficult and time consuming. Trying to get suitable subjects to test may be a problem, and then it may be hard to get any meaningful quantitative data. Quantitative data may be a necessity when comparing various alternatives for an interface, so that some conclusions can be made about the merits of each interface. Testing user performance in carrying out certain tasks can be used to compare interfaces by recording the time taken to do the tasks and the number of errors or mistakes that were made.

Aside from quantitative data the users can be asked subjective questions to see what they preferred, found most difficult etc. This can be done as a written questionnaire afterwards, or it could be done as a "think aloud" experiment where the subjects are encouraged to talk about what they are doing and thinking, and to explain why they do certain things. This can lead to very useful insights that can't be made through quantitative tests, and may not come through in written responses because of the time delay between using the interface and filling out the questionnaire.

Very detailed evaluation may be carried out by having the program itself automatically recording various details about the interaction. This can show up specific problems, say, in the way the users tend to use the mouse, or in specific problems they encounter in using different parts of the program. Logging data automatically will produce a very large amount of data (although this depends on the type and quantity of data collected) so care is needed to make efficient use of storage [13].

2.4.4 Documentation

> *Any necessary instruction or training should be needed only once; with each explanation the person should be able to say "of course", or "Yes, I see" ... If the explanation leads the person to think or say "How am I going to remember that?" the design has failed.* D Norman [20]

For any system of a reasonable size the designer can't expect the user to be able to just read the manual and remember and understand all of it [22]. So it is desirable for the program to be easy enough to operate without the user having to know everything in the manual, or to need to make frequent reference to it.

Simpson [28] gives recommendations for documentation style that can be applied to on-screen instructions as well. Instructions should be given in short sentences and with a positive construction to improve the user's comprehension. Shneiderman [27] points out that any documentation should be in simple enough language for all users to understand, specifically avoiding technical terminology. Thimbleby [30] suggests that documentation should be developed concurrently with the software, and that there should be an emphasis on minimising the documentation. Considering the casual use situation this could be extended to a goal of creating a display and interface that is so intuitive that there is no need to have any documentation or instruction at all! This would be particularly good for a science centre exhibit where a significant proportion of the users will have limited or non-existent reading ability, and often people don't read the instructions anyway. In this situation the documentation is usually one or two pages of large text and diagrams on the wall, or table, near by. Even if the instructions appear to be unnecessary, it may be best to include them because some people will be timid, not wanting to do anything unless they are told to do so through the instructions. There should, at least, be some explanation of the principles behind what the display demonstrates.

2.4.5 Help

On-line assistance is becoming more popular due to its convenience, and the features that are possible on a computerised help system such as searching, context sensitivity, skill level sensitivity, speech, and animation. However on-line help is often not used because it is hard to use, difficult to switch between the help and working context, and difficult to find appropriate and relevant information [25].

The inclusion of user assistance can't patch up a bad system, and should never be a substitute for good interface design [25]. For most systems the idea minimising documentation, which was presented in the previous section, could be extended to help facilities too. Since on-line help is effectively a form of documentation the need for it may indicate that there is scope for improving the design.

If a help facility is included then it must be easy to access — users shouldn't need help to get help — and should be tested thoroughly to ensure that it is easy to use, provides adequate explanation, and that the information can be understood by all users.

Chapter 3

Case Studies of HCI in Existing Exhibits

A few exhibits at a science centre were observed to see how well people managed to use them, and to see what problems they had. The first is a program that records and displays local weather conditions. It was not designed for this situation and was not very suitable. The second two were intended for use by children and so were more effective exhibits. The "Data Digester" collected information about visitors and was a good example of how important it is to use iterative design. The "KidPix" painting program illustrates how the model of the program can be shown so that the user knows "where" they are. Note that these comments are based only on subjective observations as no quantitative experimentation was carried out.

A science centre is a challenging environment to design for, especially for computer-based exhibits. Often computer exhibits are no more interesting than a mediocre computer game, especially when compared to the excitement of chemical reactions [3] or Van de Graaff Generators. In this situation people don't just want static things to *look* at, rather they are encouraged to get involved and interact with exhibits to make the learning experience more enjoyable and memorable. This requires robustness and durability for coping with many hours of hard use, and also good feedback so that when people first try doing something they get a reaction that encourages them to continue using the exhibit.

The time required to find out what to do and to achieve the desired results should not be too long as there are other things the visitors will want to see, and others wanting to use the display.

3.1 Weather Station

The Weather Station is a commercial program that records inside and outside temperature, wind speed and direction, and humidity. The program provides summary statistics for each of these over the last month or so as well as showing the current conditions.

Because it wasn't designed as a science exhibit there is a lot of functionality available to the user that is unsuitable for this situation. When the Weather Station was first installed the full computer keyboard was available to the user. This made it interesting for those with computer experience who wanted to cause trouble if at all possible. Data was frequently being erased and users could quit from the program leaving the display in an unsuitable state for the next person who came along.

To overcome these problems a restricted "keyboard" was provided, which restricted users to a subset of the complete program's functions. However, ~~the~~ the main problem was that it was designed for use by someone who had the time to read the manual, and not for people who wanted to be able to do something in a very short time. This was apparent in confusing terminology that needed the instruction booklet for interpretation, along with many actions that didn't have any cues to let the user know what could (and should) be done. Consequently most people would spend a little time finding they couldn't really do anything interesting or that made sense, so they would give up and find something more interesting and understandable to investigate.

Another "feature" of the program was a beep every time something invalid was tried. This made it clear that the action had no meaningful effect, however it discourages experimentation that is necessary to actually achieve anything because of the lack of instructions. More written information could be provided to explain how to operate the exhibit and to define the abbreviations and scientific terms used in the program.

One form of extra documentation that would improve the usability of the program is to provide a suitable finite-state diagram of the program structure. This could show transitions from screen to screen, along with details of those commands that were available at each state, and even an indication of how long certain actions would take if they were unusually slow. This would at least provide the user with some idea of what was happening, and would give them the knowledge needed to successfully navigate around the program. Ideally the program should be redesigned if it is to be used as a science exhibit, although showing the state diagram may make it acceptable.

As pointed out earlier, it is desirable that even systems that aren't intended for casual use are designed with the infrequent user in mind. Doing this makes using it simpler, easier, and friendlier for the frequent user, while making it accessible to the new-comer as well. If this program had been designed so that the "uninitiated" user could operate it successfully then it may have transferred to the science centre environment without much difficulty.

3.2 Data Digester

The Data Digester is an exhibit that has been set up to get feedback from visitors about their visit to the science centre, and about specific exhibits. It is essentially a questionnaire and so it involves the challenge of trying to get qualitative information from the users while making it interesting for them so that they actually use it.

Hyper-card^{TM} was used for developing the exhibit because of the flexibility provided and the speed with which the design could be implemented and modified. As problems were noticed and new ideas thought of, the display could be updated easily.

The basic model of the Data Digester is a sequence of screens, each with a simple question. Once the question has been answered, some (hopefully) interesting information is given to encourage the user in what they are doing. After finding out a few personal details the user is asked questions about their visit and a few exhibits at the centre. By asking about a few randomly chosen exhibits, information on all of them can be gleaned without pestering every user with too many questions. For this exhibit a balance is needed between questions that are considered *useful* and those that are included to keep the user interested. The input is by means of a touch screen so there is no problem with people struggling to use a mouse.

Initially there were problems with the program buffering the screen touches so that extra touches were remembered and interpreted as being made on the following screen. This happened when the program response was slow and users got impatient and started pressing the screen again[1]. The program was modified so that these extra touches were ignored.

Another problem was with the use of the touch screen. Often the program would interpret the touch as being in a quite different place to where the screen was actually touched, so incorrect responses were being recorded. Another exhibit using a touch screen had similar accuracy problems. Because of this the results will be fairly inaccurate, even for people who wanted to complete the survey correctly. In order to avoid incorrect responses being recorded, a button could be provided to let the user go back to the previous question to re-answer it if they noticed that it was recorded incorrectly. An even better solution would be to have the program allow the user to keep re-selecting a different answer until the program got it right. Then the program could wait for a special "OK" button to be pressed before continuing.

It is intended that languages other than English be made available once the content of the survey has been finalised. At present the user can select one of four languages when they start the survey but the three non-English ones respond with a message saying that it is not implemented yet. It would be better if these other options weren't available until they are fully implemented.

3.3 KidPix

KidPix is a drawing program designed specifically for children with lots of sounds and special effects to keep them interested. It is a very popular exhibit, and probably the one people spend the longest on at the science centre. Children and adults alike are amused by the sounds and animations that accompany each different drawing and erasing function.

The state diagram of the program is very complex with several modes of drawing available, and then each of these has many sub-states that are only accessible when the higher level state is the current one. The status of the program is indicated by highlighting the icons for the current state. Although this information is shown, a lot of people, especially children, seem to have trouble understanding the system completely. In particular, some of the function icons are a bit unclear, although users can easily experiment to find out what they do. The undo icon is a surprised face and people are often unsure of what it does.

[1] There was some more timely audible feedback on the buttons but initially this was not amplified and could not be heard from where the user stood.

More recently, written instructions have been put beside the exhibit giving a basic explanation of what to do. It would be a good idea to explain what the icons mean too because these can be hard to understand. The "undo" button is mentioned in the instructions but there is no picture of it so many people would not know which button is being referred to.

A mouse is used for input because of technical difficulties in using a touch screen. Often people attempt to use the mouse upside down or sideways, and they don't seem to be able to work out that rotating it would solve their difficulties in getting the pointer to move in the direction they want!

Overall this exhibit works well and keeps people amused for a long time. Slightly more detailed information could be provided to explain what the functions do, although it is mostly young children who use it so they may not be able to read it.

Chapter 4

Persistence of Vision Display

In order to gain some insight into the development of an interface for casual users, and to see how the more general principles apply to this situation, a science demonstration exhibit was developed. Over the course of the development various aspects of the design were modified and evaluated in an attempt to produce a display that was more effective, and easy to use. The first section below describes the basic design of the exhibit — the Persistence Wand. The following two sections discuss some initial evaluation, and various other design ideas. The next chapter explains some evaluation that was carried out with three variations of the exhibit.

Determining the user population characteristics was fairly easy for a science centre because we are *assuming* that a large percentage of users will be children, that they don't necessarily have any computer experience, and that their attention will need to be attracted and retained with something interesting or exciting otherwise they will give up and move on to another exhibit. However, in this setting we do not necessarily expect people to use each exhibit for particularly long, as there are plenty of other things to see and do, and hopefully there will be other people wanting to use the exhibit too.

The purpose of the exhibit that was created is to demonstrate the persistence of the human eye, and to show how a simple *raster* display works. This idea is not original, although no literature was found on it.

The basic idea is to mount several LEDs in a row and to rapidly light each one in turn, scanning through a picture, to display a simple bitmap image. The light from the individual LEDs persist in one's eye so that if the row of LEDs is quickly moved sideways across the viewers field of vision then the whole image can be seen. Figure 4.1 shows the effect for an image of an arrow. Each dot represents an LED as the wand is waved from left to right. The light grey dots indicate LEDs that are not lit for that column, while the darker ones are those that are turned on at that particular position. It is necessary to have a fairly dark environment for the image to be seen clearly.

Early feasibility experimentation found that scanning the image one pixel at a time gave a clearer effect than lighting a whole column at a time. By displaying the whole column at once, each "pixel" gets spread out sideways much more and the distinct points of light as shown in Figure 4.1 are lost. Displaying the image one pixel at a

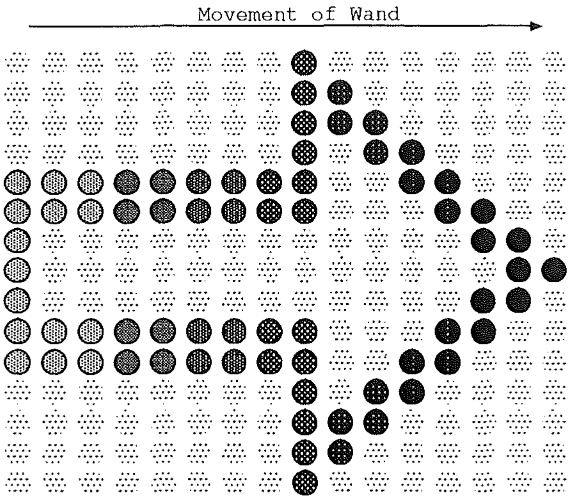

Figure 4.1: Persistence Effect From Waving the Wand

time is also a more typical method of raster display, and so it is a better technique to use in demonstrating raster principles. This method of scanning was used throughout all subsequent development.

4.1 Exhibit Design

The development of the exhibit involved an iterative design process, making improvements and enhancements as problems were observed. Several different design alternatives were considered with the final choice involving a "wand" that users wave back and forth, and a physical dial that is used to select different images to be displayed on the wand. The emphasis was on making the exhibit as easy and natural to use as possible so that everyone could get to see the persistence effect.

The following sections describe the initial version of the exhibit that was developed. The first section discusses the hardware aspect of the display, which involved interfacing the LEDs to the computer and constructing a "wand", which had the LEDs mounted in it so that they could be viewed. The second part was the actual interface that gives the user some control over the operation of the display. Other design ideas, some of which were implemented, are presented in Section 4.3.

Wand

The sideways movement that is needed to see the image can be achieved by either moving the LEDs from side to side, or keeping them still and having the user's view move. The simplest method is for the viewer to shake their head, or to glance from side to side, which spreads the image out enough for them to see it.

The alternative is to move the LEDs themselves and this could be done by some motorised or mechanical system to make them wave back and forth, either moving from side to side, or being rotated on a cylinder so that they always move the same way. Otherwise, they can be mounted in a "wand" for the viewer to wave back and forth themselves. This latter method was chosen because it makes the exhibit more *hands-on*, which is desirable for keeping interest and is really a necessity for a science centre display. Although the wand could be held still and the image viewed by these other means, the rate of scanning needed for each viewing method is quite different and so in practice they are incompatible. To see the image by glancing from side to side requires a much higher scan rate because the eye movement is a lot quicker than can be achieved by swinging the wand

A prototype wand was constructed for the purpose of testing the exhibit and the interface. Fifteen LEDs are mounted in the front and a mercury switch is located below these to detect sideways swinging. In the long run a more robust and ergonomic wand would be needed if the display is to be set up permanently.

The LEDs are controlled through the computer's parallel port using four output lines and a four to sixteen bit de-multiplexer to switch the appropriate LED on. The first output from the de-multiplexer is not connected so that an output value of "0000" results in all LEDs being off. Figure 4.2 shows the basic schematic for a 7-LED wand. To display the image the program scans down each LED in sequence, and turns it on if the corresponding bit in the bitmap image is on, otherwise a zero is output. Once a column has been completed the scanning moves on to the next column with no extra delay.

The mercury gravity switch is connected to another data line of the parallel port and is used to coordinate the display of the bitmap with the swinging of the wand. At the end of each swing the switch changes state and so it can be used to detect the ends of each swing and thus how fast the wand is being waved. Display of the picture only occurs when the wand is being swung from left to right. If the image is lit up in both directions the two images that are seen tend to interfere with each other making it harder to discern the picture because they don't coincide exactly.

One of the "pictures" used was a special one that just lit each LED in turn, and at a much slower speed. When this image is viewed the user sees interesting helix shapes due to the sinusoidal motion of the wand. At the right waving speeds the helix can be made to appear stationary, or to move slowly up or down the wand. Waving it at a slower speed can create a double helix image, like the shape of DNA.

In order to get accurate delays between each LED position a hardware timer was used to generate interrupts at regular time intervals. For the helix the best time between changing to the next LED position was about 7 milliseconds. This gives about ten columns per second. For the other pictures a range of 250 to 800 microseconds was used between each LED. This range could possibly be extended, especially at the lower end, to enable the picture to be seen at even faster waving speeds. A delay of 180 interrupts was used between detecting the switch change, and starting to display the picture.

When the wand is left idle (i.e., the switch doesn't change state for a while), the LEDs would normally remain off, but instead the image is displayed again so that

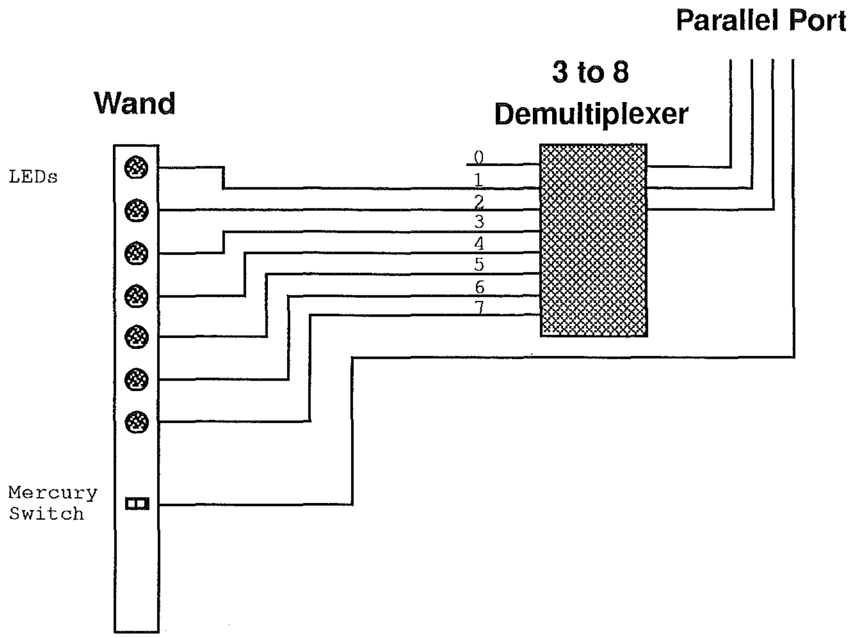

Figure 4.2: Simplified LED Wand Schematic

there is some activity to be seen on the wand. A balance needs to be found between making the delay small enough that users don't think something has gone wrong when nothing is displayed, and yet having it long enough so that the extra activity isn't confused with the main scanning.

Interface

The basic requirements of the interface were to allow selection of an image from a few choices, and control over the speed of scanning. Figure 4.3 shows the screen display that was used.

The choice of picture was made possible through six large "buttons" on the screen that each had a fifteen-by-sixteen picture on the front of them. The button that was currently selected showed the image as an array of red lights on a black background to indicate that this was the picture currently being displayed on the wand. The remaining buttons had their images rendered in black squares on a grey background so that users could see what the picture would be if they selected this button. The choice of pictures could have been better by having ones that were more familiar.

The speed control was implemented as a "slider" that could be "dragged" up and down to adjust the scanning rate. This constrains the user's selection of speed so that they can't make it go too slow or too fast to see the picture. This was made to look like a volume control on a sound mixing desk or similar. This object may not be very familiar to many people, but hopefully users can get the right impression from what

(See Fig 4.3

But what about exploration of the phenomenon. Showing it right down so that they can get to 1st principles

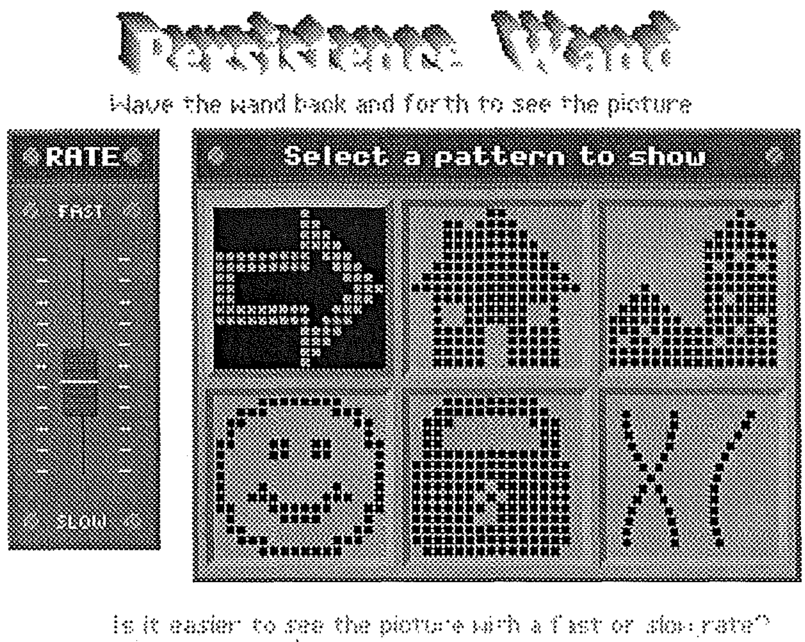

Figure 4.3: The On-screen Interface

it looks like it should do, and so use it successfully. There were some problems with this that are mentioned in section 4.3 where some alternatives are proposed.

Interaction with the interface was through the use of a mouse that enabled users to select a picture, or move the slider. The mouse was a two-button one with both buttons doing exactly the same thing. This could cause confusion for users who may tend to expect them to have different functions.

4.2 Initial Evaluation

After some casual experimentation it seemed that using a wand was going to be unsuitable. Most people who tried using it (without being given any verbal or written instruction) didn't have any idea of what they were supposed to be doing, and so they weren't successful in waving it in the right direction, let alone actually seeing the picture. However, with a quick demonstration they were soon able to use it correctly.

This led to the interesting question of *how* to convey to the user what they should be doing. It was originally thought that it would be reasonably intuitive, but it became apparent that this was not the case. An alternative design idea involved the use of mirrors and this is discussed below.

Improvements to the existing wand-based design were also considered. The original design let the user adjust the speed that the LED image was scanned so that they could experiment with different rates. However this makes it much more difficult for users to successfully see the image so it was suggested that the program should dynamically calculate an appropriate speed of scanning instead of the user having to set it. This fits in with the idea of trying to eliminate errors, and in making it easy for the user to experience success in their use of the exhibit. Doing this also eliminates the need for the user to control the speed, and the problems associated with having them understand what the "speed" means, and how to use the interface to alter it.

4.3 Alternative Designs

As mentioned in the previous section users found it difficult to know what to do with the wand, and so other methods of viewing the image were considered. Alternatives to the screen-based interface have also been proposed. These ideas are discussed in the following sections.

Creating Movement

Instead of physically moving the LEDs it is also possible to move an *image* of the LEDs. One way of achieving this is to get the user to view the LEDs in a mirror (or through a lens), and to make the mirror rotate so that the user's view of the LEDs is moved. Figure 4.4 shows how this could be done with four small mirrors mounted on a shaft so that they can be turned easily. This had been tried in the preliminary experiments and was found to work very well, and it has several advantages over the wand method. In particular there are only two directions in which the mirrors can be moved, and both "work" to produce the image as long as the speed is not too fast. A major problem with the wand is that people find it hard to work out, or see *how* it should be waved, and so they find it difficult to actually see the image. A good way needs to be found to describe the action that should be made.

Other advantages of using the mirrors include the fact that it is easier to maintain a consistent speed, it is less tiring to operate, the image can't be waved in front of the screen, and the image appears square rather than bent.

However, there is the problem of getting users to spin the mirrors at the right speed to be able to see the picture properly. With the wand it is possible to adjust the rate of display according to the shaking speed and thus keep the picture visible for a large range of swinging speeds. This could also be done for the rotating mirrors. Another solution to aid in getting the correct display speed is to motorise the mirrors or wand[1] so that the right speed of movement is obtained, although this defeats the purpose of having a hands-on interactive exhibit. A further draw-back of this scheme is that people may think that it is a trick with mirrors rather than seeing them just as a means to easily view the LEDs.

The wand is much more versatile in the movements that can be made, so the images can be purposely bent to produce interesting effects, and the wand can be waved in a circle for the helix so that a three dimensional effect is created. The helical images rely on the changing velocity of the wand to get the sine curves so this feature may

[1] If this was done the the speed of the LED scanning should be adjustable, but not the speed of the mirrors or wand!

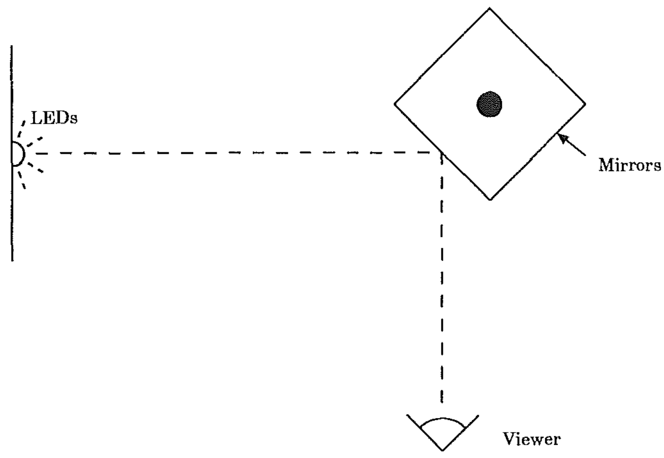

Figure 4.4: Using Mirrors to View the Image

have to be dropped for a mirror version. The same effect can be seen by oscillating the mirrors, however this is a different action to that required for the other pictures so this may cause added confusion for the user if the helix was included.

User Control

Letting users select from a few patterns to be displayed gives them scope for more interaction, and it adds a bit of variety to the exhibit. An extension to this is to let people create their own bitmaps to be shown, although this creates potential for misuse. One of the suggestions below would allow users to do this easily.

Using the slider for the speed control may be hard for people to understand and so it was suggested that this could be done with three buttons for `fast`, `medium`, or `slow`. Buttons would be easier to use, especially for small children. Alternatively, eliminating the speed controller completely and having the program do the work for the user as explained in section 4.3 below would further simplify the interface, making it even easier to use.

In the initial implementation a computer screen with mouse input was used to let the user choose between several bitmap pictures and to control the scanning speed. Using a touch-screen may make it easier for people to use the exhibit, because those unfamiliar with using a mouse often find it difficult to use at first so they aren't very successful at operating the interface. Better schemes could be used that provide other means than the screen to control the exhibit and thus hide the computer completely.

The simplest way to do this is to use a physical multi-way switch and a potentiometer, both with appropriate labelling and diagrams to describe what should be done. The simplicity of such an interface means that the user is less distracted by the controls and so can focus on the wand and achieving the persistence effect. Also having something physical to use means that tactile feedback is given, and the user doesn't have to understand the modelling of buttons on the computer screen. Because this method of input doesn't require a screen it is potentially cheaper to set up, and it may be possible to implement it entirely with a cheap stand-alone processor board, thus reducing costs even further. This form of interface was used for one of the versions of the exhibit evaluated in the next chapter.

Another idea that would give more flexibility to the user while having a more hands-on interface is to set up a fifteen by fifteen array of small switches with each one representing an LED in the picture being displayed. The switches needed for this are of the type where the current state can be controlled by the hardware, pressing it toggles it on and off, and LEDs are mounted in them to show the current state. This could allow the user to alter the picture, while seeing the current image lit up on the panel. Predefined patterns could also be selected through a multi-way switch. This too could be cheaper than a computer if it can all be done in hardware or with a cheap processor board, however it would be a rather expensive experiment if it turned out to be unsuitable.

Although these two methods could be cheaper in the long run by not using a computer, the cost of using one to experiment is virtually zero because it can be used for something else if the exhibit fails.

LED Scanning

As mentioned in section 4.2 it can be difficult to see the image if the speed of swinging doesn't correspond well with the scanning rate. If the wand is waved too slowly then the picture flashes by too quickly and isn't spread out enough for it to be seen. On the other hand, waving it too fast doesn't give time for the complete image to be scanned, so only a few columns are visible in each swing cycle.

To resolve this problem the LED interrupt routine was modified to calculate the time for each swing of the wand, and then adjust the speed of scanning to suit. This means that the user needs to be waving the wand in a reasonably regular manner for this to work properly. By automating the speed control it becomes easier to see the picture and the user doesn't have to worry about getting the speed 'just right'. The speed was adjusted so that if the swing time was the same for the next swing, then a total of 1200 interrupts would occur during that time. The choice of value for this needs some refinement.

During the idle time there needs to be some activity on the wand to indicate that the exhibit is still working. Without a screen it is especially important to do this because the LEDs provide the only feedback that anything is happening. This time can be filled by continuing to show the current image, maybe at a slower rate to show what is happening in more detail.

A further possible variation is to scan the image horizontally instead of vertically so that the wand is waved up and down rather than sideways. A number of people trying the exhibit waved it this way for a start so it seems that this direction may be more natural, although some people commented that it was more unnatural to do it this way. The major problem with doing it horizontally would be that the images would be seen upside down for those who used the opposite hand to the one the exhibit was set up for. This method of scanning was not implemented.

Chapter 5

Evaluation

This chapter discusses the Persistence of Vision display and evaluates three different versions and their suitability as science centre exhibits. The main aspects of the exhibit that were tested in the different versions were the use of a computer screen for the interface, and the effectiveness of automating the speed control. There were also two other minor differences between versions to see what effect these had. Ideally each aspect of the interface that was tested should have been done independently of the others. However, time did not permit such thorough testing.

Each exhibit was set up for a period of time in a science centre and people were observed using it. Some people were encouraged to discuss the exhibit and comment on problems or difficulties, but most were left to themselves and observed from a distance. Further casual observations were also made, mostly with the third version as this appears to be the best so far.

Observation of users in a science centre environment may be sufficient for discovering most problems with the interface, although ideally a more formal experiment could be conducted to try to get more quantitative data. This may provide further insight, although the casual observations that were made provided plenty of ideas for improvement, without being too formal. Also, feedback on what users liked or did not like, and what they could or could not understand, would be useful.

The specifics of each version are described in the section below with the differences between the versions being highlighted in Table 5.1. The second section summarises the results and draws conclusions based on the observations for each of the displays.

5.1 Exhibits

All of the setups involved the use of the wand to provide the movement, and for each one a brief instruction sheet was placed on the table giving some details of what the exhibit was about, and with a simple diagram to show how the wand should be waved. This seemed to be reasonably effective, although a more realistic illustration may have made it clearer for those who were unsure of what to do. Also, some of the instructions were found to be slightly unclear and these could be clarified by saying that the image will be seen on the *wand* and that the wand should be waved quickly.

Features	Version 1	Version 2	Version 3
Mouse & screen	✔	✔	✗
Auto-speed	✗	✔	✔
Idle time filled	✔	✗	✔
Instructions	✔	✔	✔
Colour	✔	✗	N/A

Table 5.1: Features of each experimental version

The first version evaluated was the original design with the on-screen interface and the slider for letting the user explicitly control the speed of scanning. It used colour in the interface to give clarity and interest to the presentation, and the wand continued to display the picture even when it wasn't being waved back and forth.

The second version automatically adjusted the speed of scanning according to the time taken for the previous complete swing. This meant that for most swinging speeds the image was fully visible. Other alterations included increasing the size of the instructions on the screen, removing the colour from the display, and not lighting any LEDs when the wand was idle. These changes were made to see how these three aspects of the exhibit affected usage.

The final version involved replacing the on-screen interface with a physical multi-way switch to allow selection of the different pictures. Using such a switch means that the screen and mouse are unnecessary and so the presence of a computer is *hidden*. This version also had the automatic speed implemented, and it had a simple cycling scan of the LEDs when the wand wasn't being waved.

5.2 Results

In making the observations it was unfortunate that the science centre wasn't very busy at the time that was chosen, and further observations there couldn't be made. Most of those observed used the first version so there wasn't so much feedback on the second two. Other observations were made away from the science centre mainly with the third version. Interesting observations were made, and it would appear that the third version without the screen and mouse is the best so far.

Overall the reaction to the exhibit was very positive, although several people had trouble working out what to do. Some of the reason for this could be blamed on inadequate explanation with some of the instructions not explaining what the use should do as clearly as they could have.

The wand seemed to provide a reasonably natural way to move the LEDs back and forth, although getting people to wave it in the correct way seems to be a problem that may be difficult to overcome completely. A three year old using the exhibit had difficulty in waving the wand so an adult did that for her. She was able to select pictures with the dial and identify what they were without too much trouble.

The use of the screen and mouse were found to distract the user from focussing on the 'task' of waving the wand. A significant problem encountered was that of people initially waving the wand in front of the screen. Perhaps they expected some interaction between the wand and the screen, but instead doing this makes it nearly impossible to see the LEDs against the glare of the screen.

To overcome this it may be useful to have written instruction telling the users not to wave the wand in front of the screen, although people are likely to ignore this anyway. This problem is overcome by not having the screen based interface. Brighter LEDs could be used to make the image stand out better.

Using the screen provides greater visual attraction, especially from a distance. With only the wand there is not so much to attract attention, although most people seem to work systematically around the exhibits to ensure they don't miss anything. Children will tend to be more spontaneous though, rushing around exhibits that look exciting.

Having the pictures on the screen seemed to create some confusion with users not realising that they were supposed to be seeing an image on the *wand*. The instructions could have been clearer in specifying that they were supposed to look at the wand.

The mouse caused the usual sorts of problems for some people with them attempting to use it upside down or sideways. By not using the mouse and screen these problems are avoided, and the user can focus on using the wand instead. A touch screen could be used in place of the mouse, and this would make that part of the exhibit easier to use.

Letting the user control the speed of scanning means that there is more chance they will be unsuccessful in using the exhibit. Most people who used the first version of the program didn't try altering the speed, so they were using it as it had been left by the previous person. The auto-speed feature makes it easier to see the image properly, although some users were gentle and waved the wand far too slowly to see anything. It was suggested that the instructions should tell the user to wave the wand 'quickly' or 'vigorously' so they could see the picture. This problem of people waving the wand too slowly can't be remedied by slowing the scan rate because it is the persistence of the LEDs in the eye that enables the image to be seen, and if there is too much time between the start and finish of the image the eye doesn't see the whole picture, but just parts of it flashing by.

Without lighting the LEDs when the wand is idle it is less obvious to the users that they need to look at the wand, or at which side they should look. It is especially important to have the wand displaying something when there is no screen to attract attention, although even with the screen present, the users need to be encouraged to look at and use the wand. One person using the second version, where the wand was blank when it was not being moved, couldn't figure out what to do. Having the lights going may have encouraged him to persevere a bit more.

In the third version the idle time was filled with a slow helix pattern but users found this distracting and off-putting because it was starting too soon after a lack of movement was noticed. It was also giving a slightly wrong impression of how the pictures were displayed, because the helix pattern was used rather than the current picture. If the current picture was scanned slowly during this time then this would serve as a simple slow-motion demonstration of how the wand works.

Written instructions were included with each version, although they couldn't be seen very well with the third one because it was so dark. For the first two versions the screen provided nearly adequate illumination for them to be read. Ideally the instructions could be lit from behind so that they could be seen in the dark environment.

Aside from some form of animation it would seem that the best way to indicate how to use the exhibit would be to show a long exposure photo of someone using it. This would demonstrate how the wand should be held and waved, and it would show what the expected result was too. Part of the problem for people using the exhibit was that they weren't sure what to expect, and so they didn't know whether it was worth persevering or not.

The use of colour in the first version didn't seem to have a lot of effect, with most people trying the exhibit anyway. However, the use of colour in the first display makes it look more interesting and the red LEDs, on the selected button, give a better indication of what the user should be able to see. In the third version on-screen colour is not an issue, but good use of colour in the instruction and explanation sheets could be used to clearly show the LEDs and to highlight important instructions.

Chapter 6

Conclusion

This report has discussed issues in designing human-computer interfaces particularly in relation to interfaces for casual users — those who only use a program or system for a very short time and who are likely to have no experience with computers. A few exhibits used in the setting of a science centre were described and their effectiveness as exhibits are described in chapter 3.

A simple science exhibit, the Persistence Wand, was developed so that some of these ideas could be investigated in more depth, and this is described in chapter 4. Even with all the background in HCI from chapter 2 in mind while designing the Persistence of Vision exhibit it was still necessary to make modifications and improvements to the way it operated, and to the interface. As part of this iterative design process the exhibit was tested with various people in an attempt to see how it could be made easier for them to use. There is still more work that could be done to refine the instructions provided, improve the exact scan rates and timings that are used, and to develop a final version of the actual wand.

The following is a list of some principles that can be applied to the design of interfaces. Particular focus is given to science displays, or similar systems, where novice and inexperienced people will use the program.

- Know the user. Knowledge of the nature and type of user that will use a particular system is very important. If the specific needs of the potential user population isn't met then they will find the system frustrating and difficult to use. The novice user requires a very simple interface, and if the system is only for novice or casual users then the design can be much simpler.

- Hide the computer. When there is a computer involved it can be distracting and threatening for many novice users. If the presence of a computer or microprocessor can be hidden behind suitable control panels and input devices, then there is increased *transparency* at the interface and users will be able to focus on the purpose of the system rather than the interface.

- Careful presentation. Colour, graphics, sound, and animation can all be used to make a display more interesting and clearer to understand. They can also make it more confusing and cluttered than necessary, so care is needed to make good use of these aspects of the interface. Cushman [9] provides various useful guidelines and quantitative principles for many aspects of the human-computer interface.

- Be consistent. The interface should provide consistent interpretation of user input, and where appropriate, make use of existing standards in interface design. Of course, if better methods or metaphors are suitable to the situation then they could be used, as long as doing this doesn't cause unnecessary confusion.

- Use appropriate input devices. For each situation there will be different input requirements and these need to be assessed to determine what is best. For casual users a touch-screen is preferred over mice, but if the interface can be reduced to a few physical switches, or proportional controllers, then the distraction of the computer will be minimised and the system made easier to use.

- Positive and immediate feedback. Messages should be phrased positively and clearly so that the user knows what is happening and is informed in a friendly manner. Any feedback that is provided needs to be acknowledged immediately, and if this is done then *no* response is often the best way to indicate an invalid action.

- Minimise the potential for errors. There are many methods that can be used in an interface to limit the user's input rather than letting them enter something that is invalid. By using these techniques the interface is made easier and less frustrating to use, and the user can be more productive.

- Keep the model simple. The finite state model of the system should be made as simple as possible, and it should always be clear to the user what state the system is in. If there is the slightest bit of complexity then it is suggested that the state model should be shown, or at least made known, in some way.

- Use suitable metaphors. The use of metaphors in an interface can greatly aid the user in understanding how components of the interface work. Ideas and concepts from the world that are familiar to the user should be employed to help convey the meaning and semantics of objects and operations in the interface.

- Use iterative design. The designer must be prepared to make both major and minor modifications to the design as the need is discovered in the testing. Time needs to be allowed for this, and it is helpful to use a development environment where it is easy to make the necessary modifications.

- Test with "real" users. Thorough testing is necessary so that deficiencies and problems in the design can be discovered, and then improvements can be made. This involves observing people from the potential user community using the system — designers are useless for testing because they know too much about the system.

- Minimise documentation. If something can be explained briefly then there is a greater chance that the user will be able to understand it easily. At the extreme, something that is very easy to use needs no instruction. A door that needs a label ("push", "pull" or "slide") is not as easy to use as it could be.

If these ideas and guidelines are considered and applied in the design of interfaces for casual users then it is much more likely that the product will be easy, and even fun to use. Designing for complete novices does not have to be restricted to science exhibits or public information systems. If these principles can be applied to more complex applications too, then they may also be made more usable, and people might even be able to enjoy using them.

If all products and systems were developed with the intent of serving the user in their needs, and applying principles from human-computer interaction research, then the whole population would benefit greatly through systems that are much easier and more productive to use.

Bibliography

[1] ARMSTRONG, M. P., AND DENSHAM, P. J. Cartographic Visualization and User Interfaces in Spatial Decision Support Systems. In *GIS/LIS '91 proceedings* (1991), vol. 1, pp. 321–330.

[2] BAECKER, R., AND SMALL, I. Animation at the Interface. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 251–267.

[3] BELL, T. Computer Science for the Uninterested: Designing Displays for a Science Centre. *Computers in NZ Schools* (1992), 40–46.

[4] BENNETT, K. B., AND FLACH, J. M. Graphical Displays: Implications for Divided Attention, Focused Attention and Problem Solving. *Human Factors 34*, 5 (Oct. 1992), 513–534. Special Isuue: Visual Displays.

[5] BUXTON, B. HCI and the Inadequacies of Direct Manipulation Systems. *SIGCHI 25*, 1 (Jan. 1993), 21–22.

[6] BUXTON, W. There's More to Interaction Than Meets the Eye: Some Issues in Manual Input. In *User Centred System Design*, D. E. Norman and S. W. Draper, Eds. 1986, ch. 15, pp. 319–337.

[7] CAMPBELL, J., AND STEIDLER, F. The Strässle Approach to a modern GIS. In *EGIS '92* (1992), vol. 1, pp. 564–573.

[8] CRAWFORD, C. Lessons from Computer Game Design. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 103–111.

[9] CUSHMAN, W. H., AND ROSENBERG, D. J. *Human Factors in Product Design*, vol. 14 of *Advances in Human Factors/Ergonomics*. Elsevier, Amsterdam, 1991.

[10] DUMAIS, S. T., AND LANDAUER, T. K. Psychological Investigations of Natural Terminology for Command and Query Languages. In *Directions in Human/Computer Interaction*, A. Badre and B. Shneiderman, Eds. Ablex Publishing Company, 1981, ch. 5, pp. 95–110.

[11] EBERTS, R. E. Augmented Display for Problem Solving. In *Human-computer Interaction: Proceedings of the First USA-Japan Conference on Human-Computer Interaction* (1984), G. Salvendy, Ed., pp. 261–265.

[12] ERIKSON, T. D. Working with Interface Metaphors. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 65–73.

[13] Ford, G. J. Tirohia Te Mahi: An Empirical Study of Children's Use of a Maori Language Word Processor. Master's thesis, 1993.

[14] Freundshuh, S. M., and Gould, M. D. Empirical User Testing for Validation of GIS design. In *GIS/LIS '91 proceedings* (1991), vol. 2, pp. 922–931.

[15] Knapp, B. G., Moses, F. L., and Gellman, L. H. Information Highlighting on Complex Displays. In *Directions in Human/Computer Interaction*, A. Badre and B. Shneiderman, Eds. Ablex Publishing Company, 1981, ch. 10, pp. 195–215.

[16] Leung, Y. K. A Comparative Study of Graphical User Interfaces for Topological Maps. Tech. Rep. A:92/1, School of Mathematical and Informatin Sciences, Massey University, Palmerston North, New Zealand, Dec. 1992.

[17] Lewis, C., and Norman, D. A. Designing for Error. In *User Centred System Design*, D. E. Norman and S. W. Draper, Eds. 1986, ch. 20, pp. 411–432.

[18] McMaster, R. B., and Mark, D. M. The Design of a Graphical User Interface for Knowledge Acquisition in Cartographic Generalization. In *GIS/LIS '91 proceedings* (1991), vol. 1, pp. 311–320.

[19] Michaelis, P. R., and Wiggins, R. H. A Human Factors Engineer's Introduction to Speech Synthesis. In *Directions in Human/Computer Interaction*, A. Badre and B. Shneiderman, Eds. Ablex Publishing Company, 1981, ch. 8.

[20] Norman, D. A. *The Psychology of Everyday Things*. Basic Books Inc Publishers, New York, 1988.

[21] Norman, D. A. Why Interfaces Don't Work. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 209–219.

[22] Preece, J., Ed. *A Guide to Usability. Human Factors in Computing*. Addison-Wesley, 1993.

[23] Rheingold, H. An Interview with Don Norman. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 209–219.

[24] Salomon, G. New Uses for Color. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 269–278.

[25] Sellen, A., and Nicol, A. Building User-Centered On-line Help. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 143–153.

[26] Shneiderman, B. System Message Design: Guidelines and Experimental Results. In *Directions in Human/Computer Interaction*, A. Badre and B. Shneiderman, Eds. Ablex Publishing Company, 1981, ch. 3, pp. 55–78.

[27] Shneiderman, B. *Designing the User Interface*, 2 ed. Addison-Wesley, 1992.

[28] Simpson, H., and Casey, S. M. *Developing Effective User Documentation*. McGraw Hill, 1988.

[29] SUTCLIFFE, A. *Human-Computer Interface Design.* Macmillan Education, 1988.

[30] THIMBLEBY, H. *User Interface Design.* Addison-Wesley, New York, 1992.

[31] TOGNAZZINI, B. Consistency. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 1990, pp. 75–77.

[32] TUFTE, E. R. *The Visual Display of Quantitative Information.* Graphics Press, 1983.

[33] VIRZI, R. A. Refining the Test Phase of Usability Evaluation: How Many Subjects is Enough? *Human Factors 34*, 4 (Aug. 1992), 457–468.