CHAPTER 1-

Worked exercise *Identify the goals and operators involved in the problem 'delete the second paragraph of the document' on a word processor. Now use a word processor to delete a paragraph and note your actions, goals and subgoals. How well did they match your earlier description?*

Answer Assume you have a document open and you are at some arbitrary position within it. You also need to decide which operators are available and what their preconditions and results are. Based on an imaginary word processor we assume the following operators (you may wish to use your own WP package):

Operator	Precondition	Result
delete_paragraph	Cursor at start of paragraph	Paragraph deleted
move_to_paragraph	Cursor anywhere in document	Cursor moves to start of next paragraph (except where there is no next paragraph when no effect)
move_to_start	Cursor anywhere in document	Cursor at start of document

Goal: *delete second paragraph in document*
Looking at the operators an obvious one to resolve this goal is delete_paragraph which has the precondition 'cursor at start of paragraph'. We therefore have a new subgoal: *move_to_paragraph*. The precondition is 'cursor anywhere in document' (which we can meet) but we want the second paragraph so we must initially be in the first.

We set up a new subgoal, move_to_start, with precondition 'cursor anywhere in document' and result 'cursor at start of document'. We can then apply *move_to_paragraph* and finally *delete_paragraph*.

We assume some knowledge here (that the second paragraph is the paragraph after the first one).

CHAPTER 2

OCR - Optical character recognition or optical character reader

Paper-based interaction

Paper is principally seen as an output medium. You type in some text, format it, print it and read it. The idea of the paperless office was to remove the paper from the write–read loop entirely, but it didn't fundamentally challenge its place in the cycle as an output medium. However, this view of paper as output has changed as OCR technology has improved and scanners become commonplace.

Workers at Xerox Palo Alto Research Center (also known as Xerox PARC) capitalized on this by using paper as a medium of interaction with computer systems [195]. A special identifying mark is printed onto forms and similar output. The printed forms may have check boxes or areas for writing numbers or (in block capitals!) words. The form can then be scanned back in. The system reads the identifying mark and thereby knows what sort of paper form it is dealing with. It doesn't have to use OCR on the printed text of the form as it printed it, but can detect the check boxes that have been filled in and even recognize the text that has been written. The identifying mark the researchers used is composed of backward and forward slashes, '\' and '/', and is called a *glyph*. An alternative would have been to use bar codes, but the slashes were found to fax and scan more reliably. The research version of this system was known as XAX, but it is now marketed as Xerox PaperWorks.

One application of this technology is mail order catalogs. The order form is printed with a glyph. When completed, forms can simply be collected into bundles and scanned in batches, generating orders automatically. If the customer faxes an order the fax-receiving software recognizes the glyph and the order is processed without ever being handled at the company end. Such a *paper user interface* may involve no screens or keyboards whatsoever.

Some types of paper now have identifying marks micro-printed like a form of textured watermark. This can be used both to identify the piece of paper (as the glyph does), and to identify the location on the paper. If this book were printed on such paper it would be possible to point at a word or diagram with a special pen-like device and have it work out what page you are on and where you are pointing and thus take you to appropriate web materials . . . perhaps the fourth edition . . .

It is paradoxical that Xerox PARC, where much of the driving work behind current 'mouse and window' computer interfaces began, has also developed this totally non-screen and non-mouse paradigm. However, the common principle behind each is the novel and appropriate use of different media for graceful interaction.

Cartography is the study and practice of making maps.

Worked exercise *What input and output devices would you use for the following systems? For each, compare and contrast alternatives, and if appropriate indicate why the conventional keyboard, mouse and CRT screen may be less suitable.*

(a) portable word processor
(b) tourist information system
(c) tractor-mounted crop-spraying controller

(d) air traffic control system
(e) worldwide personal communications system
(f) digital cartographic system.

Answer In the later exercise on basic architecture (see Section 2.8.6), we focus on 'typical' systems, whereas here the emphasis is on the diversity of different devices needed for specialized purposes. You can 'collect' devices – watch out for shop tills, bank tellers, taxi meters, lift buttons, domestic appliances, etc.

(a) Portable word processor
The determining factors are size, weight and battery power. However, remember the purpose: this is a word processor not an address book or even a data entry device.
(i) LCD screen – low-power requirement
(ii) trackball or stylus for pointing
(iii) real keyboard – you can't word process without a reasonable keyboard and stylus handwriting recognition is not good enough
(iv) small, low-power bubble-jet printer – although not always necessary, this makes the package stand alone. It is probably not so necessary that the printer has a large battery capacity as printing can probably wait until a power point is found.

(b) Tourist information system
This is likely to be in a public place. Most users will only visit the system once, so

(b) Tourist information system
This is likely to be in a public place. Most users will only visit the system once, so the information and mode of interaction must be immediately obvious.
(i) touchscreen only – easy and direct interaction for first-time users (see also Chapter 3)
(ii) NO mice or styluses – in a public place they wouldn't stay long!

(c) Tractor-mounted crop-spraying controller
A hostile environment with plenty of mud and chemicals. Requires numerical input for flow rates, etc., but probably no text
(i) touch-sensitive keypad – ordinary keypads would get blocked up
(ii) small dedicated LED display (LCDs often can't be read in sunlight and large screens are fragile)
(iii) again no mice or styluses – they would get lost.

(d) Air traffic control system
The emphasis is on immediately available information and rapid interaction. The controller cannot afford to spend time searching for information; all frequently used information must be readily available.
(i) several specialized displays – including overlays of electronic information on radar
(ii) light pen or stylus – high-precision direct interaction
(iii) keyboard – for occasional text input, but consider making it fold out of the way.

(e) Worldwide personal communications system
Basically a super mobile phone! If it is to be kept on hand all the time it must be very light and pocket sized. However, to be a 'communications' system one would imagine that it should also act as a personal address/telephone book, etc.

(i) standard telephone keypad – the most frequent use
(ii) small dedicated LCD display – low power, specialized functions
(iii) possibly stylus for interaction – it allows relatively rich interaction with the address book software, but little space
(iv) a 'docking' facility – the system itself will be too small for a full-sized keyboard(!), but you won't want to enter in all your addresses and telephone numbers by stylus!

(f) Digital cartographic system
This calls for very high-precision input and output facilities. It is similar to CAD in terms of the screen facilities and printing, but in addition will require specialized data capture.
(i) large high-resolution color VDU (20 inch or bigger) – these tend to be enormously big (from back to front). LCD screens, although promising far thinner displays in the long term, cannot at present be made large enough
(ii) digitizing tablet – for tracing data on existing paper maps. It could also double up as a pointing device for some interaction
(iii) possibly thumbwheels – for detailed pointing and positioning tasks
(iv) large-format printer – indeed very large: an A2 or A1 plotter at minimum.

Worked exercise *What is the basic architecture of a computer system?*

Answer In an HCI context, you should be assessing the architecture from the point of view of the user. The material for this question is scattered throughout the chapter. Look too at personal computer magazines, where adverts and articles will give you some idea of typical capabilities . . . and costs. They may also raise some questions: just what is the difference to the user between an 8 ms and a 10 ms disk drive?

The example answer below gives the general style, although more detail would be expected of a full answer. In particular, you need to develop a feel for capacities either as ball-park figures or in terms of typical capabilities (seconds of video, pages of text).

Example
The basic architecture of a computer system consists of the computer itself (with associated memory), input and output devices for user interaction and various forms of hard-copy devices. (Note, the 'computer science' answer regards output to the user and output to a printer as essentially equivalent. This is not an acceptable user-centered view.)

A typical configuration of user input–output devices would be a screen with a keyboard for typing text and a mouse for pointing and positioning. Depending on circumstance, different pointing devices may be used such as a stylus (for more direct interaction) or a touchpad (especially on portable computers).

The computer itself can be considered as composed of some processing element and memory. The memory is itself divided into short-term memory which is lost when the machine is turned off and permanent memory which persists.

Worked exercise *How do you think new, fast, high-density memory devices and quick processors have influenced recent developments in HCI? Do they make systems any easier to use? Do they expand the range of applications of computer systems?*

Answer Arguably it is not so much the increase in computer power as the decrease in the cost of that power which has had the most profound effect. Because 'ordinary' users have powerful machines on their desktops it has become possible to view that power as available for the interface rather than hoarded for number-crunching applications.

Modern graphical interaction consumes vast amounts of processing power and would have been completely impossible only a few years ago. There is an extent to which systems have to run faster to stay still, in that as screen size, resolution and color range increase, so does the necessary processing power to maintain the 'same' interaction. However, this extra processing is not really producing the same effect; screen quality is still a major block on effective interaction.

The increase in RAM means that larger programs can be written, effectively allowing the programmer 'elbow room'. This is used in two ways: to allow extra functionality and to support easier interaction. Whether the former really improves usability is debatable – unused functionality is a good marketing point, but is of no benefit to the user. The ease of use of a system is often determined by a host of small features, such as the

appropriate choice of default options. These features make the interface seem 'simple', but make the program very complex . . . and large. Certainly the availability of elbow room, both in terms of memory and processing power, has made such features possible.

The increase in both short-term (RAM) and long-term (disks and optical storage) memory has also removed many of the arbitrary limits in systems: it is possible to edit documents of virtually unlimited size and to treat the computer (suitably backed up) as one's primary information repository.

Some whole new application areas have become possible because of advances in memory and processing. Most applications of multimedia including voice recognition and online storage and capture of video and audio, require enormous amounts of processing and/or memory. In particular, large magnetic and optical storage devices have been the key to electronic document storage whereby all paper documents are scanned and stored within a computer system. In some contexts such systems have completely replaced paper-based filing cabinets.

CHAPTER 6

Worked exercise *What is the distinction between a process-oriented and a structure-oriented design rationale technique? Would you classify psychological design rationale as process or structure oriented? Why?*

Answer The distinction between a process- and structure-oriented design rationale resides in what information the design rationale attempts to capture. Process-oriented design rationale is interested in recording an historically accurate description of a design team making some decision on a particular issue for the design. In this sense, process-oriented design rationale becomes an activity concurrent with the rest of the design

process. Structure-oriented design rationale is less interested in preserving the historical evolution of the design. Rather, it is more interested in providing the conclusions of the design activity, so it can be done in a post hoc and reflective manner after the fact.

The purpose of psychological design rationale is to support the task–artifact cycle. Here, the tasks that the users perform are changed by the systems on which they perform the tasks. A psychological design rationale proceeds by having the designers of the system record what they believe are the tasks that the system should support and then building the system to support the tasks. The designers suggest scenarios for the tasks which will be used to observe new users of the system. Observations of the users provide the information needed for the actual design rationale of that version of the system. The consequences of the design's assumptions about the important tasks are then gauged against the actual use in an attempt to justify the design or suggest improvements.

Psychological design rationale is mainly a process-oriented approach. The activity of a claims analysis is precisely about capturing what the designers assumed about the system at one point in time and how those assumptions compared with actual use. Therefore, the history of the psychological design rationale is important. The discipline involved in performing a psychological design rationale requires designers to perform the claims analysis during the actual design activity, and not as post hoc reconstruction.

CHAPTER 7

Worked exercise *Look at some of the principles outlined in this section, and use one or two to provide a usability specification (see Chapter 6, Section 6.3) for an electronic meetings diary or calendar. First identify some of the tasks that would be performed by a user trying to keep track of future meetings, and then complete the usability specification assuming that the electronic system will be replacing a paper-based system. What assumptions do you have to make about the user and the electronic diary in order to create a reasonable usability specification?*

Answer This exercise could be easily extended to a small project which would involve the design of such an electronic diary or calendar. The purpose of this smaller usability engineering exercise is to show how usability goals can be formulated early on to drive the design activity. We will select two of the usability principles from this chapter, which will serve as attributes for separate usability specifications.

In the first example, we will consider the interaction principle of guessability, which concerns how easy it is for new users to perform tasks initially. The measuring concept will be how long it takes a new user, without any instruction on the new system, to enter his first appointment in the diary. A sample usability specification is given below.

Attribute:	Guessability
Measuring concept:	Ease of first use of system without training
Measuring method:	Time to create first entry in diary
Now level:	30 seconds on paper-based system
Worst case:	1 minute
Planned level:	45 seconds
Best case:	30 seconds (equivalent to now)

The values in this usability specification might seem a little surprising at first, since we are saying that the best case is only equivalent to the currently achievable now level. The point in this example is that the new system is replacing a very familiar paper and pencil system which requires very little training. The objective of this system is not so much to improve guessability but to preserve it. Earlier, we discussed that the worst case level should not usually be worse than the now level, but we are hoping for this product to improve overall functionality of the system. The user will be able to do more things with the electronic diary than he could with the conventional system. As a result, we worry less about improving its guessability. Perhaps we could have been more ambitious in setting the best case value by considering the potential for voice input or other exotic input techniques that would make entry faster than writing.

As another example, we want to support the task migratability of the system. A frequent sort of task for a diary is to schedule weekly meetings. The conventional system would require the user to make an explicit entry for the meeting each week – the task of the scheduling is the responsibility of the user. In the new system, we want to allow the user to push the responsibility of scheduling over to the system, so that the user need only indicate the desire to have a meeting scheduled for a certain time each week and the system will take care of entering the meeting at all of the appropriate times. The task of scheduling has thus migrated over to the system. The usability specification for this example follows.

Attribute:	Task migratability
Measuring concept:	Scheduling a weekly meeting
Measuring method:	Time it takes to enter a weekly meeting appointment
Now level:	(Time to schedule one appointment) × (Number of weeks)
Worst case:	Time to schedule two appointments
Planned level:	1.5 × (Time to schedule one appointment)
Best case:	Time to schedule one appointment

In this specification, we have indicated that the now level is equivalent to the time it takes to schedule each appointment separately. The worst, planned and best case levels are all targeted at some proportion of the time it takes to schedule just a single appointment – a dramatic improvement. The difference between the worst, planned and best case levels is the amount of overhead it will take to indicate that a single appointment is to be considered an example to be repeated at the weekly level.

What are the assumptions we have to make in order to arrive at such a usability specification? One of the problems with usability specifications, discussed earlier, is that they sometimes require quite specific information about the design. For example, had we set one of our measuring methods to count keystrokes or mouse clicks, we would

have had to start making assumptions about the method of interaction that the system would allow. Had we tried to set a usability specification concerning the browsing of the diary, we would have had to start making assumptions about the layout of the calendar (monthly, weekly, daily) in order to make our estimates specific enough to measure. In the examples we have provided above, we have tried to stay as abstract as possible, so that the usability specifications could be of use as early in the design life cycle as possible. A consequence of this abstractness, particularly evident in the second example, is that we run the risk in the usability specification of setting goals that may be completely unrealistic, though well intentioned. If the usability specification were to be used as a contract with the customer, such speculation could spell real trouble for the designer.

Worked exercise *Look up and report back guidelines for the use of color. Be able to state the empirical psychological evidence that supports the guidelines. Do the guidelines conflict with any other known guidelines? Which principles of interaction do they support?*

Answer There are many examples of guidelines for the use of color in the literature. Here are three good sources:

- C. Marlin Brown, *Human–Computer Interface Design Guidelines*, Ablex, 1988.
- Deborah J. Mayhew, *Principles and Guidelines in Software User Interface Design*, Prentice Hall, 1992.
- Sun Microsystems, Inc., *OpenLook Graphical User Interface Application Style Guidelines*, Addison-Wesley, 1990.

Taking an example from Mayhew, we have the following design guideline for the use of color as an informational cue for the user (for example, to inform the user that a string of text is a warning or error message):

> Do not use color without some other redundant cue.

Mayhew provides three reasons which empirically support this guideline:

1. Color may not be available on all machines on which the system is to be implemented. Therefore, if use of color is the only means to convey some important information to the user, then that information will be lost in a monochrome (no color) system. Redundant color coding will allow for portability across different computing platforms.
2. Empirical evidence shows that 8% of the (general) male population and 0.4% of the female population has some color deficiency, so they cannot accurately recognize or distinguish between various colors. Again, if color is the only means for conveying some information, this significant portion of the user population will be slighted.
3. It has been shown that redundant color coding enhances user performance

This guideline supports several of the principles discussed in this chapter:

Substitutivity The system is able to substitute color-coded information and other means (for example, text, sound) to represent some important information. We could turn the argument around and suggest that the user be able to provide color input (by selecting from a palette menu) or other forms of input to provide relevant information to the system.

Observability This principle is all about the system being able to provide the user with enough information about its internal state to assist his task. Relying strictly on color-coded information, as pointed out above, could reduce the observability of a system for some users.

Synthesis If a change in color is used to indicate the changing status of some system entity (perhaps a change in temperature above a threshold value is signalled by an icon becoming red), those who cannot detect the change in color would be deprived of this information. Synthesis is about supporting the user's ability to detect such significant changes, especially when they are a result of previous user actions.

There is no evidence of existing guidelines that this particular guideline for color violates.

Another example of a color guideline (found in all three of the above references) is the demand to consider cultural information in the selection of particular colors. For example, Mayhew states that western cultures tend to interpret green to mean go or safe; red to mean stop, on, hot or emergency; and blue to mean cold or off. Using color to suggest these kinds of meanings is in support of the familiarity principle within learnability. However, in other cultures different meanings may be associated with these colors, as we saw in Chapter 3, and consistent use of color (another guideline) might lead to confusion. Hence, strict adherence to this guideline would suggest a violation of the consistency of color application guideline. However, if consistency is applied relative to the meaning of the color (as opposed to its actual color), this guideline would not have to conflict.