4.5 Design principles

Over the years many principles of good interactive system design have been developed. Don Norman in his book *The Design of Everyday Things* (Norman, 1998) provides several, as does Jacob Nielsen in *Usability Engineering* (Nielsen, 1993). However, the level of abstraction provided by different people at different times is sometimes rather inconsistent and confusing. Design principles can be very broad or they can be more specific. There are also good design principles that derive from psychology, such as 'minimize memory load', i.e. do not expect people to remember too much. (We discuss many of these principles in Chapters 12 and 13 on interface design.) Apple, Microsoft and Google all provide user interface design guidelines for the development of products that run on their platforms.

The application of design principles has led to established design guidelines and patterns of interaction in certain circumstances such as the 'Undo' command in a Windows application, the 'Back' button on a website or the greying-out of inappropriate options on menus.

→ Memory and attention are discussed in Chapter 21

Design principles can guide the designer during the design process and can be used to evaluate and critique prototype design ideas. Our list of high-level design principles, put together from Norman, Nielsen and others, is shown below. All the principles interact in complex ways, affecting each other, sometimes conflicting with each other and sometimes enhancing each other. But they help to orientate the designer to key features of good design and sensitize the designer to important issues.

For ease of memorizing and use we have grouped them into three main categories – learnability, effectiveness and accommodation – but these groupings are not rigid. Systems should be learnable, effective *and* accommodating.

- Principles 1–4 are concerned with access, ease of learning and remembering (learnability).
- Principles 5–7 are concerned with ease of use, and principles 8 and 9 with safety (effectiveness).
- Principles 10–12 are concerned with accommodating differences between people and respecting those differences (accommodation).

Designing interactive systems from a human-centred perspective is concerned with the following.

Helping people access, learn and remember the system

1 *Visibility.* Try to ensure that things are visible so that people can see what functions are available and what the system is currently doing. This is an important part of the psychological principle that it is easier to recognize things than to have to recall them. If it is not possible to make it visible, make it observable. Consider making things 'visible' through the use of sound and touch.

2 *Consistency.* Be consistent in the use of design features and be consistent with similar systems and standard ways of working. Consistency can be something of a slippery concept (see the Further thoughts box). Both conceptual and physical consistency are important.

→ Metaphors are discussed in Section 9.3

3 *Familiarity.* Use language and symbols that the intended audience will be familiar with. Where this is not possible because the concepts are quite different from those people know about, provide a suitable metaphor to help them transfer similar and related knowledge from a more familiar domain.

4 *Affordance*. Design things so it is clear what they are for; for example, make buttons look like push buttons so people will press them. Affordance refers to the properties that things have (or are perceived to have) and how these relate to how the things could be used. Buttons afford pressing, chairs afford sitting on, and Post-it notes afford writing a message on and sticking next to something else. Affordances are culturally determined.

Giving them the sense of being in control, knowing what to do and how to do it

5 *Navigation*. Provide support to enable people to move around the parts of the system: maps, directional signs and information signs.

6 *Control*. Make it clear who or what is in control and allow people to take control. Control is enhanced if there is a clear, logical mapping between controls and the effect that they have. Also make clear the relationship between what the system does and what will happen in the world outside the system.

7 *Feedback*. Rapidly feed back information from the system to people so that they know what effect their actions have had. Constant and consistent feedback will enhance the feeling of control.

Safely and securely

8 *Recovery*. Enable recovery from actions, particularly mistakes and errors, quickly and effectively.

9 *Constraints*. Provide constraints so that people do not try to do things that are inappropriate. In particular, people should be prevented from making serious errors through properly constraining allowable actions and seeking confirmation of dangerous operations.

In a way that suits them

10 *Flexibility*. Allow multiple ways of doing things so as to accommodate people with different levels of experience and interest in the system. Provide people with the opportunity to change the way things look or behave so that they can personalize the system.

11 *Style*. Designs should be stylish and attractive.

12 *Conviviality*. Interactive systems should be polite, friendly and generally pleasant. Nothing ruins the experience of using an interactive system more than an aggressive message or an abrupt interruption. Design for politeness (see Box 4.4). Conviviality also suggests joining in and using interactive technologies to connect and support people.

FURTHER THOUGHTS

Consistency

Consistency is a slippery concept because consistency is always relative. A design will be consistent with respect to some things but may be inconsistent with respect to others. There are also times when to be inconsistent is a good thing because it draws people's attention to something that is important. The difference between conceptual consistency and physical consistency is important. Conceptual consistency is about ensuring the mappings are consistent, that the conceptual model remains clear. This involves being consistent both internally to the system and externally as the system relates to things outside it. Physical consistency is ensuring consistent behaviours and consistent use of colours, names, layout and so on.

One famous example of the difficulty of maintaining conceptual consistency in a design comes from the design of the Xerox Star interface (described in Smith *et al.*, 1982).

→

To print a document, the document was dragged onto a printer icon. This was consistent with the overall style. The question then arose as to what to do with it after it had been printed. The options considered were (1) the system deletes the icon from the desktop, or (2) the system does not delete the icon, but (a) replaces it on the desktop in its previous location, (b) places it at an arbitrary location on the desktop, or (c) leaves it on the printer for the user to deal with. Discuss!

Kellogg (1989) quotes the designers as saying that in this example the trade-off was between the external consistency of not deleting the icon, as it behaved more like a real-world object (a photocopier), against the internal consistency of behaving like other actions in the interface, such as dragging the icon to the wastebasket or to a folder icon. They opted for option 2a. Whether designers would do that nowadays, when more people are much more familiar with these types of interface, is another matter.

BOX 4.4

Polite software

Alan Cooper (1999) argues that if we want people to like our software we should design it to behave like a likeable person. Drawing on work by Reeves and Nass (1996), who found that people interacting with new media were treating the media like a person ('The Media Equation'), they argue that the essentials of polite behaviour are quality, quantity, relevance and clarity. Cooper continues with his list of characteristics:

Polite software:

is interested in me
is deferential to me
is forthcoming
has common sense
anticipates my needs
is responsive
gives instant gratification
is taciturn about its personal problems
is well informed
is perceptive
is self-confident
stays focused
is fudge-able
is trustworthy

Design principles in action

In Part III of this book we look at design in a number of specific contexts including the Web, cooperative systems, mobile computing and ubiquitous computing systems. Specific design issues and principles for those contexts are discussed there. There are also related issues discussed in Chapter 12 on interface design. Here we look at some general examples of the design principles in action.

The computer 'desktop' is likely to remain with us for some time, with its familiar combination of windows, icons, menus and pointer, called a WIMP interface. This form of interaction – the graphical user interface (GUI) – is as ubiquitous as information and communication technologies are becoming and appears on handhelds and other mobile devices as well as on desktop computers.

Designing for GUIS, is still dominated primarily by issues of usability. In particular, the key issue is **consistency**. There are clear guidelines for issues such as menu layout, ordering, dialogue boxes and use of the other 'widgets' associated with graphical user interfaces. There are standards for providing **constraints** such as greying out items on a menu that are not relevant at a particular point. A toolkit, or a design environment such as Visual Basic, will probably be used that will help to ensure that the design conforms to an overall style.

Screen design is a key issue in such environments and attention needs to be paid to the layout of objects on a screen. Avoiding clutter will help to ensure **visibility**. Attention needs to be paid to the use of appropriate, non-clashing colours and the careful layout of information using tables, graphs or text as appropriate. However, on mobile applications visibility is very difficult to achieve.

Often in the design of GUI applications, the designer can talk to the actual future stakeholders of the system and find out what they want and how they refer to things. This will help the designer to ensure that **familiar** language is used and that the design follows any organizational conventions. It can be fitted in with preferred ways of working. Participatory design techniques – involving people closely in the design process – can be used, and stakeholders can participate in the design process through workshops, meetings and evaluation of design ideas. Documentation and training can be given.

A good design will ensure that there is easy error **recovery** by providing warning signs for drastic actions such as 'Are you sure you want to destroy the database?'. A good example of designing for recovery is the Undo command.

Affordances are provided by following GUI design guidelines. People will expect to see a menu at the top of the screen and will expect the menu items to be displayed when the header is clicked on. Items that are not greyed out will afford selecting. The various 'widgets' such as check boxes, radio buttons and text entry boxes should afford selecting because people familiar with the standards will know what to expect. However, care needs to be taken to ensure that opportunities are easily and correctly perceived. On mobile devices the physical buttons afford pressing, but because of the limited screen space the same button has to do different things at different times. This leads to problems of **consistency**.

Menus are also the main form of **navigation** in GUI applications. People move around the application by selecting items from menus and then by following dialogue structures. Many applications make use of 'wizards'. These provide step-by-step instructions for undertaking a sequence of operations, allowing users to go forwards and backwards to ensure that all steps are completed.

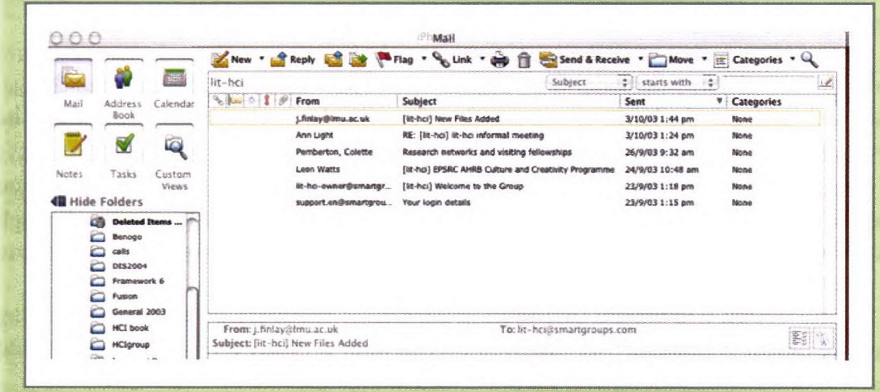
Control is usually left in the hands of the users. They have to initiate actions, although some features that provide security are undertaken automatically. Many applications, for example, automatically save people's work to help with recovery if mistakes are made. **Feedback** is provided in a variety of ways. A 'bee' symbol or an 'egg timer' symbol is used to indicate that the system is busy doing something. Counters and progress bars are used to indicate how much of an operation is complete. Feedback can be provided through sound, such as a beep when a message is received on an e-mail system or a sound to indicate that a file has been safely saved.

Flexibility is provided with things such as shortcut keys, allowing more expert users to use combinations of keyboard controls in place of using menus to initiate commands and navigate through the system. Many windows applications allow the user to set their own preferences, to configure features such as the navigation bars and menu items and to disable features that are not often used.

In terms of **style** and **conviviality**, GUI applications are rather limited as they should remain within the standard design guidelines (although Windows 8 is certainly more aesthetic than previous versions). Error messages are one area where the designer can move towards a more convivial design by thinking hard about the words used on the messages. However, all too frequently messages appear very abruptly and interrupt people unnecessarily.

Challenge 4.4

Look at Figure 4.8, an example of a typical 'windows'-type application. Critique the design from the perspective of the general design principles and from design for OS X in particular.



From	Subject	Sent	Categories
j.finlay@lmu.ac.uk	[lit-hci] New Files Added	3/10/03 1:44 pm	None
Ann Light	RE: [lit-hci] lit-hci informal meeting	3/10/03 1:24 pm	None
Pemberton, Colette	Research networks and visiting fellowships	26/9/03 9:32 am	None
Leon Watts	[lit-hci] EPSRC AHRB Culture and Creativity Programme	24/9/03 10:48 am	None
lit-hci-owner@smartgr...	[lit-hci] Welcome to the Group	23/9/03 1:18 pm	None
support.en@smartgrou...	Your login details	23/9/03 1:15 pm	None

From: j.finlay@lmu.ac.uk To: lit-hci@smartgroups.com
Subject: [lit-hci] New Files Added

Figure 4.8 'Entourage' on the Mac

(Source: Screenshot frame reprinted by permission of Microsoft Corporation)

→ See Section 14.3 on Information architecture for websites

Navigation is a central issue in website design. Even if a site is well focused, it will soon get large and so issues of how to move around a website become important. Designers need to provide support to enable people to discover the structure and content of the site and to find their way to a particular part of the site. Information architecture is an established area of study devoted to designing websites.

A key feature of **consistency** is the use of standard Web features such as a blue underline for showing a link. Many sites confuse people by not making links sufficiently visible and distinguishable from other text on the site. **Flexibility** of navigation can be enabled by providing alternatives for people; different routes though the site and having a variety of links. Having a site map will **afford** people getting oriented.

Issues of **recovery**, **feedback** and **control** figure most highly in shopping sites. There are often long pauses when processing things such as a payment transaction. Feedback is critical here and statements such as 'this action may take 45 seconds to complete' are used to persuade people not to do anything while the transaction is processed. However, there is no way of enforcing **constraints** in these circumstances.

→ See Section 5.2 on engagement

Conviviality can be provided by allowing people to join in, to support and create communities. Unlike standard GUI applications, websites can easily connect people with one another. **Style** is also key to websites and offers the most opportunities for designers to demonstrate their creative flair. The use of animation, video and other design features can really develop a whole sense of engagement with the site.

Summary and key points

Good design is about usability. It is about ensuring that systems are accessible to all and that designs are acceptable for the people and contexts in which they will be used. Designers need to evaluate their designs with people and involve people in the design