Interface metaphors and conceptual models

(as a container for discarded objects). One of these is to represent the place where disk icons are put in order to eject the corresponding disk from the disk drive. This implies that one has to ‘throw away’ a disk in order to retrieve it. Do you think that these contradictions can cause conceptual problems, and if so can you suggest another way of representing this operation at the interface?

A counter-argument against conceptual confusion is to treat such problems as actual learning experiences. For example, Carroll et al. (1988a) describe how they observed a user trying to tear off a sheet of paper from a stationery pad icon by dragging the cursor across an icon representing the stationery pad. The interface had not been designed to allow this action; sheets of paper could only be selected from a menu option. However, what Carroll et al. noted was that the invalid action enabled the user to understand better the difference between menu-based and mouse-based operations. Accordingly, she developed a more elaborate mental model of the interface.

Paradoxically, it could be the case that the more unexpected and the more bizarre the interface metaphor is, the more likely the user will develop a better understanding of the system. By carrying out inappropriate actions the user is enlightened as to what are the appropriate and permissible operations. Provided the user is able to experience the ‘ah-ha’ phenomenon relatively easily, and not be humiliated or frustrated in the process, such a strategy may be very effective. However, where the interface metaphor contravenes deep-rooted expectations (cf. the voice mail system and the central heating control models discussed in Chapter 6) users may find it difficult to switch their models and adapt to the new way of understanding how an object works. Furthermore, if the interface metaphor elicits inappropriate actions that have undesirable consequences, such as unexpectedly deleting files, the users may be reluctant to experiment further with the interface and hence never learn the full functionality of the system.

A counter-argument against conceptual confusion is to treat such problems as actual learning experiences. For example, Carroll et al. (1988a) describe how they observed a user trying to tear off a sheet of paper from a stationery pad icon by dragging the cursor across an icon representing the stationery pad. The interface had not been designed to allow this action; sheets of paper could only be selected from a menu option. However, what Carroll et al. noted was that the invalid action enabled the user to understand better the difference between menu-based and mouse-based operations. Accordingly, she developed a more elaborate mental model of the interface.

Paradoxically, it could be the case that the more unexpected and the more bizarre the interface metaphor is, the more likely the user will develop a better understanding of the system. By carrying out inappropriate actions the user is enlightened as to what are the appropriate and permissible operations. Provided the user is able to experience the ‘ah-ha’ phenomenon relatively easily, and not be humiliated or frustrated in the process, such a strategy may be very effective. However, where the interface metaphor contravenes deep-rooted expectations (cf. the voice mail system and the central heating control models discussed in Chapter 6) users may find it difficult to switch their models and adapt to the new way of understanding how an object works. Furthermore, if the interface metaphor elicits inappropriate actions that have undesirable consequences, such as unexpectedly deleting files, the users may be reluctant to experiment further with the interface and hence never learn the full functionality of the system.
system, designers can begin to have a better understanding of what metaphors are most appropriate (Erikson, 1990).

### 7.3 Classification of interface metaphors for applications

The desktop metaphor and its composites have been the most successful and pervasive of all interface metaphors. There are other metaphors, however, which have been developed for applications other than information systems (see Chapter 22). Table 7.1 presents some examples with their associated applications.

An important consideration when searching for interface metaphors is the appropriate effect. This is the subjective and emotional impact that different graphical representations can convey (Verplank, 1988). The kinds of metaphors like spreadsheets and multi-agents may be very appealing to adults working in office environments, but may be inappropriate for schoolchildren. Imaginary characters like demons and wizards might be far more attractive. Some sketching techniques for helping you to explore different metaphors by brainstorming are discussed in Chapter 22.

<table>
<thead>
<tr>
<th>Application area</th>
<th>Metaphor</th>
<th>Familiar knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating environment</td>
<td>The desktop</td>
<td>Office tasks, file management</td>
</tr>
<tr>
<td>Spreadsheets</td>
<td>Ledger sheet</td>
<td>Columnar tables</td>
</tr>
<tr>
<td>Object-oriented environments</td>
<td>Physical world</td>
<td>Real-world behaviour</td>
</tr>
<tr>
<td>Hypertext</td>
<td>Notecards</td>
<td>Flexible organization of structured text</td>
</tr>
<tr>
<td>Learning environments</td>
<td>Travel</td>
<td>Tours, guides, navigation</td>
</tr>
<tr>
<td>File storage</td>
<td>Piles</td>
<td>Categorizing objects in terms of urgency, projects and so on</td>
</tr>
<tr>
<td>Multimedia environments</td>
<td>Rooms (each associated with a different medium/task)</td>
<td>Spatial structure of buildings</td>
</tr>
<tr>
<td>Computer supported cooperative work</td>
<td>Multi-agents</td>
<td>Travel agents, butlers and other serving roles</td>
</tr>
</tbody>
</table>

### 7.4 Ubiquitous computing

**Ubiquitous computing** is a term coined by Weiser (1991) for ‘invisibly enhancing the world that already exists’ (p. 61). The ultimate aim of ubiquitous computing is to make the interface metaphor invisible to the user in the same way as computer
systems are invisible in home appliances, such as the VCR, the microwave oven and the washing machine. Just as we use these tools unconsciously and effortlessly to accomplish our everyday tasks (with the exception of the VCR, whose interface has generally been all too visible and difficult to use) it is envisaged that ubiquitous computing systems will be effortless to use.

An example of a ubiquitous system is the ‘tab’ which is a tiny networked computer that is intended to be worn by the user, shown in Figure 7.2. The tab or ‘active badge’ can identify itself throughout a building, making it possible to keep track of the person it is attached to. This can be very useful for buildings with high security areas in so far as the person wearing the tab can enter secure areas without having to remember a password and then key it in, as the tab does it automatically. Among other opportunities such automaticity affords are: on entering a room people can be greeted by name, telephone calls can be automatically forwarded and computer terminals can change to the customization specified by the person sitting at it.

Negreponte (1989) also discusses the idea of invisible computers that intercommunicate with each other with the purpose of doing all the chores in life.
Instead of you having to decide that you need to buy some more milk, or that the heating needs turning up or the trash needs emptying, a ‘society of objects’ in the form of virtual butlers, secretaries and housekeepers will organize and manage everything for you! However, a problem with such extensive anthropomorphism is that users may assume that the system is more intelligent than it is. When the virtual agents fail to behave as expected users may get frustrated. There is also the danger that designers could mislead users in undesirable ways.

**Question 7.4**

*What is ubiquitous computing?*

### 7.5 Conceptual models

**Conceptual models** is the generic term that describes the various ways in which systems are understood by different people. Primarily these consist of (i) the way users conceptualize and understand the system and (ii) the way designers conceptualize and view the system.

As we said in Chapter 6, whether interacting with devices, machines, computers, people or the physical world, people use their prior knowledge to develop mental models to enable them to understand and predict their behaviour. A highly successful approach in interface design is to capitalize on users’ existing knowledge and the use of metaphors. However, the problem confronting designers who follow this approach is finding a suitable metaphor. The aim for designers is to help users to develop accurate mental models of the system. As Donald Norman (1986, p. 46) puts it, ‘The problem is to design the system so that, first, it follows a consistent, coherent conceptualisation — a design model — and, second, so that the user can develop a mental model of that system — a user model — consistent with the design model’.

### Users’ models, design models and the system image

An important consideration of conceptual models is the relationship between designers’ models — the design model — and users’ mental models — the user’s model shown in Figure 7.3. As most designers work in teams, it is more accurate to consider the design model as the product of a collection of individuals rather than the outcome of any one individual. Ideally, the user model should map onto the design model. That way the users will be able to use the system’s full capability as intended by the designer. However, in the real world this does not often happen. More often, users only develop a partial mental model of the design model. Their understanding and ability to use the system, therefore, is limited. Another problem is that the design model may be inappropriate for what the user wants to achieve. In this situation the
Figure 7.3 The design model, the user’s model and the system image (Norman and Draper, 1986, p. 46).

users are forced to develop a mental model that is unfamiliar to them. A mismatch can also arise if the design model is ambiguous, inconsistent or obscure.

Generally, the way users get to find out about the design model is through the interface, its behaviour and the documentation. Collectively, these are called the **system image** as shown in Figure 7.3. A large part of the accessible system image comprises the physical interface (that is, the knobs and dials or images on a screen). It is important to bear in mind, however, that the system image also includes the system’s behaviour, that is, the way it is used (for example by pressing keys, moving a mouse, and so on). The sequences of operations required in using the system (such as commands or menu selections) and the resulting events are all part of the system’s image, and users learn not just from looking at the system, but also from their experience of using it.

If the system image is not able to convey to the users the design model in a clear and obvious way, then it is likely that the users will develop incorrect mental models. Consequently, they will experience great difficulties in understanding the system, using the system and knowing what to do when the system doesn’t behave in the way they assumed it would. Much frustration, time-wasting and error-making can result. In Section 13.7, we shall extend these concepts further in relation to the design of direct manipulation systems. In particular, we shall consider what happens when the user’s model does not match the designers’ model well, creating ‘gulfs’ between the two, and how designers can help to prevent this happening.

**Question 7.5**

(a) What is the difference between a design model and a user model?
(b) What is the system image?
(c) Why do mismatches occur between the design model and the user’s model?
Key points

- Verbal metaphors are analogies based on familiar knowledge, spontaneously elicited or used in written or spoken instructions, which help users begin to understand new systems.
- Interface metaphors combine a familiar domain with the system structure to make a concrete system image.
- Composite interface metaphors are a combination of multiple and partial models of familiar domains.
- There are several kinds of interface metaphors, although the most common one is the desktop.
- Users and designers both have conceptual models of the system; Norman has called these the user and the design model, respectively.
- A main goal in interface design is to develop a system image that maps the design model onto the user model.
- Ubiquitous computing systems have invisible interfaces, are interconnected and intended to be effortless to use.
- A conceptual model provides a framework from which to consider how to design appropriate interface metaphors.

Further reading

**Metaphors and language**


A wealth of books are available in the linguistic, psychological and philosophical literature, but this is one of the most readable, lively and thought-provoking books, which has now become a ‘classic’.

**Interface metaphors**


These two chapters provide several useful suggestions on how to generate and design interface metaphors.
Conceptual models