Part I

Essentials of designing interactive systems

Introduction to Part I

Our goal is to design interactive systems that are enjoyable to use, that do useful things and that enhance the lives of the people who use them. We want our interactive systems to be accessible, usable and engaging. In order to achieve this we believe that the design of such systems should be human-centred. That is, designers need to put people rather than technology at the centre of their design process. Unfortunately, the design of interactive systems and products in the past has not always had a good record of considering the people who use them. Many systems have been designed by programmers who use computers every working day. Many designers are young males. Many designers have been playing computer games for years. This means that they forget just how difficult and obscure some of their designs can be to people who have not had these experiences.

In the days of the Web, issues of usability are critical to e-commerce. Before the immediacy of e-commerce, usability problems were discovered only after purchase. If you bought a nice-looking smartphone and brought it home only to find it was difficult to use, you could not take it back! The shop would say that it delivers its functions; all you had to do was to learn how to operate it properly. On the Web, customers look at usability first. If the system is hard to use, or if they do not understand it, they will go somewhere else to make their purchase. People are learning that systems do not have to be hard to use and are becoming more critical about the design of other products too.

This first part of the book provides a guide to the essence of the human-centred design of interactive systems. Chapter 1 focuses on the main elements of interactive systems design. It considers the nature of design, the features of interactive systems and what it means to be human-centred. The chapter provides a brief history of human–computer interaction and interaction design and a glimpse of the future, before focusing on why designing interactive systems is important. Chapter 2 introduces the key components of interaction – people, activities, contexts and technologies (PACT). This proves to be an insightful construct not just for understanding the breadth of interaction design, but also for doing design. The chapter describes and illustrates a first design method: PACT analysis.

Alongside this view we need to consider the products we are designing: what they will do, how they will do it and what information content they will manipulate. In Chapter 3 we look at the processes involved in designing interactive systems. We see why the evaluation of ideas is central to the process if we are going to be focused on people: 'being human-centred'. The requirements for products, early designs and prototypes of systems all need to be evaluated to ensure that they meet the needs of the people who will use them. But people will make use of technologies in many different contexts, to undertake different activities. The chapter introduces key abstractions for helping designers in their tasks: personas and scenarios. We give examples of personas and offer practical advice on how they can be developed and used. The chapter goes on to provide a whole scenario-based design method, providing advanced treatment of this important idea.

In Chapter 4 we look at principles of design: how to ensure systems are accessible, usable and acceptable. As interactive systems become increasingly embedded in society, they stop being a luxury. Accessibility is about ensuring that the benefits of interaction design are available to all. Another key concept in interaction design that has long been the central focus of human–computer interaction (HCI) is usability. Chapter 4 provides a detailed consideration of usability and acceptability. Finally the chapter provides some high-level design guidelines that will help designers ensure that designs are accessible and usable.

When people use the devices we have designed, what do they feel? Do they have a sense of satisfaction, enjoyment and engagement? Chapter 5 looks at these issues and at aesthetics and designing for pleasure. Once again this serves to illustrate the wide scope of interactive systems design. The chapter also includes some discussion of service design as increasingly designers need to design services as well as products. The final chapter is an extended case study of a design, showing how and why decisions were made and illustrating many of the ideas developed in the first five chapters.

After studying this part you should understand the essential features of designing interactive systems. In particular:

- What interactive systems design is
- Who is involved
- What is involved
- How to develop systems that are human-centred
- Principles of interactive systems design to ensure systems are usable and engaging.

Case studies

The concepts and ideas are illustrated throughout through a number of case studies. Chapter 1 introduces several modern devices that have made a big impact on the world of interaction design. Chapter 2 uses the development of a swipe-card system to illustrate the PACT method. Chapter 3 introduces the MP3 player case study. This involves the development of an MP3 function for the Home Information Centre (HIC) which is itself the focus of the extended case study in Chapter 6. Both the MP3 and the overall HIC case studies are also used in Part II.

Teaching and learning

With some supplementary material showing examples, following up on the Web links and further reading and doing some assessed exercises, the material in this part would make an ideal introductory course on human-computer interaction or interaction design. The list of topics covered in this part is shown below, each of which could take 10–15 hours of study to reach a good general level of understanding, or 3–5 hours for a basic appreciation of the issues. Of course, each topic could be the subject of extensive and in-depth study.

Topic 1.1	Overview of designing interactive systems	Chapter 1
Topic 1.2	Characteristics of people	Section 2.2
Topic 1.3	Activities, contexts and technologies	Sections 2.3–2.5
Topic 1.4	Doing a PACT analysis	Sections 2.1, 2.6
Topic 1.5	The design process	Section 3.1
Topic 1.6	Personas and scenarios	Section 3.2
Topic 1.7	Scenario-based design	Sections 3.3–3.4
Topic 1.8	Accessibility	Sections 4.1–4.2
Topic 1.9	Usability and acceptability	Sections 4.3–4.4
Topic 1.10	Interaction design principles	Section 4.5

Chapter 1

Designing interactive systems: a fusion of skills

Contents

Aims

Designing interactive systems is concerned with developing high-quality interactive systems, products and services that fit with people and their ways of living. Computing and communication devices are embedded in all sorts of everyday devices such as washing machines and televisions, ticket machines and jewellery. No self-respecting exhibition, museum or library is without its interactive component. We carry and wear technologies that are far more powerful than the computers of just a few years ago. There are websites, on-line communities, 'apps' for mobile phones and tablets and all manner of other interactive devices and services that need developing. Interactive systems design is about all this.

In this chapter we explore the width and breadth of designing interactive systems. After studying this chapter you should be able to:

- Understand the concepts underlying the design of interactive systems
- Understand why being human-centred is important in design
- Understand the historical background to the subject
- Understand the skills and knowledge that the designer of interactive systems needs to draw upon.

1.1 The variety of interactive systems

Designing interactive systems is concerned with many different types of product. It is about designing software systems that will run on a computer at work. It is about designing websites, games, interactive products such as MP3 players, digital cameras and applications for tablet PCs (personal computers). It is about designing whole environments in which phones, tablets, laptop computers, digital projectors and other devices and services communicate with one another and through which people interact with one another. It is about designing interactive systems, products and services for the home, for work or to support communities.

Here are some examples of recent interactive products and systems.

Example 1: The iPhone

In 2007 Apple Inc. changed the face of mobile technologies when they introduced the iPhone (Figure 1.1). The iPhone had a carefully crafted, purpose-designed interface to make use of the finger as the input device. It had a revolutionary touch-sensitive screen that allowed for multi-touch input. This facilitated new interaction techniques such as pinching an image and drawing it in to make it smaller, or pinching and moving the fingers out to make an image larger. Many mobile devices and larger screen systems have now adopted this technology, but the iPhone started it. The iPhone also included sensors that could register how the phone was being held and whether it was vertical, horizontal or sloping. This allows for other novel interaction methods. For example, the display would automatically adjust from portrait style to landscape. In 2008 the 'app store' was launched, turning the iPhone into an open platform for developers to design and produce their own software. Combined with the iTunes delivery service, this turned the iPhone into a versatile, multimedia device with hundreds of thousands of applications, from sophisticated games to trivial pieces of entertainment to useful information applications. This created new experiences and new services for a new set of customers that has now spread to many other devices running the Android operating system (from Google) or Windows (from Microsoft). The most recent iPhone has introduced a speech recognition system called Siri that allows people to call or text their friends, enter appointments in a calendar or search the Web just by speaking into the phone.

Figure 1.1 iPhone
(Source: Hannah Gal/Science Photo Library)

Example 2: The Nintendo Wii

Also in 2007 Nintendo introduced the Wii (Figure 1.2). The Wii was a revolutionary new games concept that used infra-red sensors attached to a TV or other display device to track a wand that transmitted infra-red signals. The new system could, therefore, register different gestures such as a 'bowling' action, a 'tennis shot' action or a host of other movements. The notion of computer games changed radically, from a young person shooting at imaginary monsters, or driving imaginary cars, to a family-wide entertainment. When the Wii Fit was introduced it appealed to a new audience of people wanting to keep fit at home. In 2011 Microsoft introduced their Kinnect system that combined infra-red detection and cameras so that users could interact with software using gestures with no need for a wand. Originally aimed at people playing games on the Xbox games machine, the Kinnect was quickly adapted to work with any software that could make use of its application program interface (API).

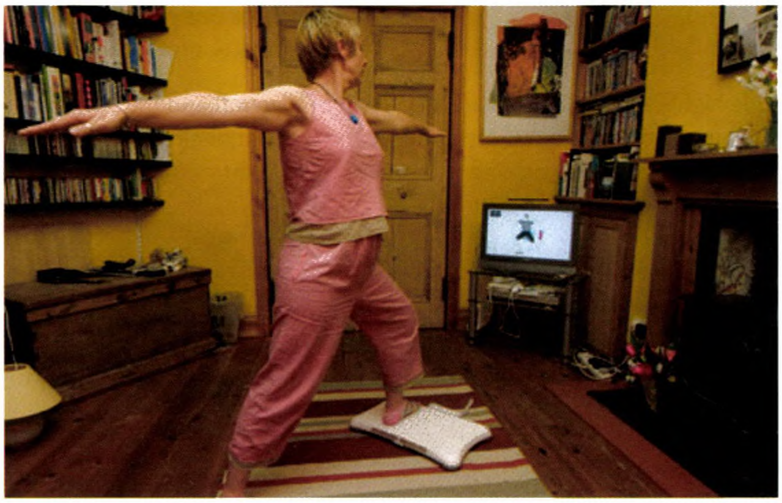

Figure 1.2 Wii Fit
(Source: Keith Morris/Alamy Images)

Example 3: Virtual worlds

Second Life (Figure 1.3) is a huge on-line community populated by animated virtual people (called avatars). It consists of thousands of simulated buildings, parks, seasides, factories, universities and everything else one could find in the real world (and much else besides). People create avatars to represent themselves in this virtual world. They can determine their size, shape, gender and what they want to wear. They are controlled by their creators using the Internet, interacting with other avatars, and visiting virtual places. Other examples of virtual

Artificial life

FURTHER THOUGHTS

Artificial life (often abbreviated to 'Alife') is a branch of artificial intelligence (AI), the discipline that looks at whether intelligent software systems can be built and at the nature of intelligence itself. The tradition in AI has been to represent knowledge and behaviours through rules and rigid structures. Alife tries instead to represent higher-level features of the things in an environment, such as the goals that a creature has and the needs that it must satisfy. The actual behaviour of the artificial creatures is then more unpredictable and evolves in the environment. Increasingly, characters in computer games are using Alife techniques.

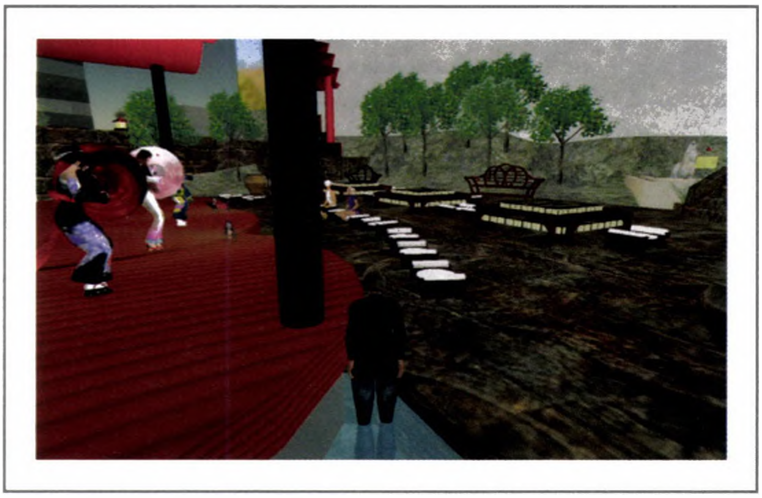

Figure 1.3 Second Life
(Source: http://secondlife.com, Linden Lab)

Figure 1.4 Sony Vita
(Source: Kiyoshi Ota/Getty Images)

worlds include highly popular games such as World of Warcraft and the Sony Home environment that is played on their Playstation platform and Vita handheld device (Figure 1.4). Many of these games include playing on-line with others, a key part of the social side of designing interactive systems.

Example 4: i Robo-Q domestic toy robot

The i Robo-Q domestic toy robot is an example of the new children's toys that are increasingly available (Figure 1.5). Toys are using all manner of new technologies to enhance the experiences of children at play. They use robotics, voice input and output, and a variety of sensors to provide novel and engaging interactions.

Example 5: Facebook

Facebook (Figure 1.6) is a highly popular website that allows people to keep in contact with their friends. Known as social networking sites, there are many similar systems around. Facebook is the most popular with nearly 1 billion users worldwide. Facebook is increasingly becoming an important platform for a wide variety of activities and it allows people to add applications (apps) in a similar way to the Apple and Android platforms. People can store and share digital

Figure 1.5 i Robo-Q domestic toy robot
(Source: Getty Images/ChinaFotoPress)

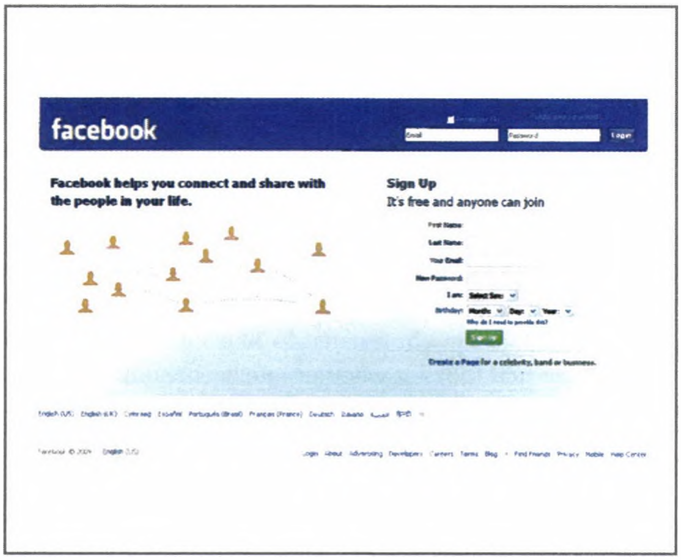

Figure 1.6 Facebook
(Source: Facebook, Inc.)

photos, write notes to each other and get regular updates about what their friends are doing. Facebook will probably have its own mobile handset soon as it moves from being just a website into being an important platform for the delivery of all sorts of interactive systems.

Summary

These five examples of interactive systems capture many of the features that the interactive systems designer has to work with. The designer of interactive systems needs to understand the possibilities that exist for new forms of interaction, with fixed devices or mobiles, for people on their own or for connecting people to each other through text messages or through animation and video. It is a fascinating area to work in.

Challenge 1.1

Find five interactive products or systems that you use – perhaps a coffee machine, a cellular phone, a fairground ride, a TV remote control, a computer game and a website. Write down what it is that you like about each of them and what it is that you do not like. Think about the whole experience and not just the functions. Think about the content that each provides: Is it what you want? Is it fun to use?

If possible, find a friend or colleague to discuss the issues. Criticism and design are social activities that are best done with others. What do you agree on? What do you disagree on? Why?

1.2 The concerns of interactive systems design

The design of interactive systems covers a very wide range of activities. Sometimes designers will be working on both the hardware and the software for a system, in which case the term 'product design' seems to be most appropriate to describe what they are doing. Sometimes the designer will be producing a piece of software to run on a computer, on a programmable device or over the Internet. In these cases the terms 'system design' and 'service design' seem more appropriate. We switch between these expressions as appropriate. However, the key concerns of the designer of interactive systems are:

- *Design*. What is design and how should you do it?
- *Technologies*. These are the interactive systems, products, devices and components themselves.
- *People* who will use the systems and whose lives would we like to make better through our designs?
- *Activities and contexts*. What do people want to do? What are the contexts within which those activities take place?

Design

> *What is design? It's where you stand with a foot in two worlds – the world of technology and the world of people and human purposes – and you try to bring the two together.*
>
> Mitch Kapor in Winograd (1996), p. 1

The term 'design' refers both to the creative process of specifying something new and to the representations that are produced during the process. So, for example, to design a website a designer will produce and evaluate various designs, such as a design of the page layout, a design of the colour scheme, a design for the graphics and a design of the overall structure. In a different field of design, an architect produces sketches and outlines and discusses these with the client before formalizing a design in the form of a blueprint.

Design is rarely a straightforward process and typically involves much iteration and exploration of both requirements (what the system is meant to do and the qualities it should have) and design solutions. There are many definitions of 'design'. Most definitions recognize that *both* problem and solution need to *evolve* during the design process; rarely can you completely specify something before some design work has been done.

One thing that is useful is to distinguish the amount of formality associated with a design:

- At one end of a spectrum is engineering design (such as the design of a bridge, a car or a building) where scientific principles and technical specifications are employed to produce formal models before construction starts.
- At the other end of this spectrum is creative or artistic design where innovation, imagination and conceptual ideas are the key ingredients.
- Somewhere in the middle lies 'design as craft' which draws upon both engineering and creative approaches.

Most design involves aspects of all of these. A fashion designer needs to know about people and fabrics, an interior designer also needs to know about paints, lighting and so on, and a jewellery designer needs to know about precious stones and the properties of metals such as gold and silver. The famous design commentator Donald Schön has described design as a 'conversation with materials', by which he means that in any type of design, designers must understand the nature of the materials that they are working with. Design works with, and shapes, a medium; in our case this medium consists of interactive systems. Others emphasize that design is a conscious, social activity and that much design is often undertaken in a design team.

People and technologies

Interactive system is the term we use to describe the technologies that interactive system designers work with. This term is intended to cover components, devices, products and software systems that are primarily concerned with processing information. Interactive systems are things that deal with the transmission, display, storage or transformation of information that people can perceive. They are devices and systems that respond dynamically to people's actions.

This definition is intended to exclude things such as tables, chairs and doors (since they do not process information) but to include things such as:

- Mobile phones (since they transmit, store and transform information)
- Websites (since they store and display information and respond to people's actions)
- Computer game controllers.

Increasingly, interactive components are being included in all manner of other products (such as clothes, buildings and cameras).

A fundamental challenge for interactive systems designers is to deal with the fact that people and interactive systems are different (see Box 1.1). Of course we take the people-centred view, but many designers still take the machine-centred view because it is quicker and easier for them, though not for the person who finishes up using the product. Another difference between people and machines is that we speak different languages. People express their desires and feelings in terms of what they want to do or how they would like things to be (their goals). Machines need to be given strict instructions.

BOX 1.1

Machine- and people-centred views

View	People are	Machines are
Machine-centred	Vague Disorganized Distractible Emotional Illogical	Precise Orderly Undistractible Unemotional Logical
People-centred	Creative Compliant Attentive to change Resourceful Able to make flexible decisions based on content	Dumb Rigid Insensitive to change Unimaginative Constrained to make consistent decisions

Source: Adapted from Norman (1993), p. 224

The interface

The interface to an interactive system, also called the user interface (UI), is all those parts of the system with which people come into contact, physically, perceptually and conceptually:

- Physically we might interact with a device by pressing buttons or moving levers and the interactive device might respond by providing feedback through the pressure of the button or lever.
- Perceptually the device displays things on a screen which we can see, or makes noises which we can hear.
- Conceptually we interact with a device by trying to work out what it does and what we should be doing. The device provides messages and other displays which are designed to help us do this.

The interface needs to provide some mechanisms so that people can provide instructions and enter data into the system: 'input'. It also needs to provide some mechanisms for the system to tell people what is happening by providing feedback and mechanisms for displaying the content: 'output'. This content might be in the form of information, pictures, movies, animations and so on. Figure 1.7 shows a variety of interfaces.

→ Chapter 2 discusses input and output devices in more detail

Challenge 1.2

Look at the pictures in Figure 1.7. What does the interface to (a) the remote control, (b) the microwave, (c) the palmtop computer or (d) the Xbox controller consist of?

Designing interactive systems is not just a question of designing interfaces, however. The whole human–computer interaction needs to be considered, as does the human–human interaction that is often enabled through the systems. Increasingly, interactive systems consist of many interconnected devices, some worn by people, some embedded in the fabric of buildings, some carried. Interactive systems designers are concerned with connecting people through devices and systems; they need to consider the whole environment they are creating.

Figure 1.7 Various user interfaces: remote control; microwave; palmtop; and Xbox controller

(Source: (a) Fujitsu; (b) © D. Hurst/Alamy Images; (c) Gareth Boden/Pearson Education Ltd. (d) Microsoft Limited)

Being human-centred

Interactive systems design is ultimately about creating interactive experiences for people. Being human-centred is about putting people first; it is about designing interactive systems to support people and for people to enjoy. Being human-centred is about:

- Thinking about what people want to do rather than what the technology can do
- Designing new ways to connect people with people
- Involving people in the design process
- Designing for diversity.

BOX 1.2

The evolving nature of interactive systems design

The primary discipline contributing to being human-centred in design is human-computer interaction (HCI). HCI arose during the early 1980s, evolving into a subject 'concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them' (ACM SIGCHI, 1992, http://old.sigchi.org/cdg/index.html).

HCI drew on cognitive psychology for its theoretical base and on software engineering for its design approach. During the 1990s the closely related area of Computer Supported Cooperative Work (CSCW) focused on technology support for cooperative activities and brought with it another theoretical base that included sociology and anthropological methods. At the same time, designers in many different fields found that they had to deal with interactive products and components, and in 1989 the first computer-related design course was established at the Royal College of Art in London. In America the designers at Apple were putting their ideas together in a book called *The Art of Human-Computer Interface Design* (Laurel, 1990a) and a meeting at Stanford University in 1992 resulted in the book *Bringing Design to Software* (Winograd, 1996). By the mid-2000s interaction design was firmly established as a discipline in its own right with the first textbooks on interaction design coming out (including the first edition of this book) and leading designers contributing their own insights. All this – coupled with the phenomenal changes in computing and communication technologies during the same period – has brought us to where we are today: a dynamic mix of ideas, approaches and philosophies applied to the design of interactive systems and products.

This book is about human-centred interactive systems design. It is about human–computer interaction (HCI) and interaction design in the twenty-first century.

1.3 Being digital

In 1995 Nicholas Negroponte, head of the Massachusetts Institute of Technology's 'Media Lab', wrote a book called *Being Digital* in which he explored the significance of an era in which we change atoms for bits. We live in a digital age, when all manner of devices represent things using binary digits (bits). The significance of being digital is that bits are transformable, transmittable and storable using digital technologies. Consider the following scenario.

In the morning you get woken up by a digital alarm clock which automatically turns on the radio. To change the radio channel you might press a button that searches for a strong signal. You pick up your mobile, cellular phone and check for messages. You might go to your computer and download a personalized newspaper into a tablet device. As you leave the house you set the security alarm. In the car you adjust the heating, use the radio and attend to the various warning and information symbols that detect whether doors are open, or seat belts are buckled. Arriving at the station, you scan your season ticket through the car parking machine, get a train ticket from the ticket machine and get money from an automated teller machine (ATM). On the train you read the newspaper on your tablet, scrolling through text using your finger. Arriving at your office, you log on to the computer network, check e-mail, use various computer packages, browse the Web and perhaps listen to an Internet radio station broadcasting from another country. You have a video link with colleagues in other cities and perhaps work together on a shared document. During the day you use a coffee machine, make calls on the cellphone, check names and numbers in the address book, download a new ringing tone, photograph a beautiful plant that you see at lunchtime and video the swans on the river. You upload these to your social networking website where they are automatically tagged with the location and time they were taken, and with the names of people whose faces the software recognised. Arriving home, you open the garage doors automatically by keying a number on your phone and in the evening you spend an hour or so on the games machine, watch TV and program the set top box to record a late-night show.

This is the world we are living in and the world that designers of interactive systems are designing for. The huge range of interactions that we engage in and the interfaces that we use offer an exciting if daunting challenge. Moreover, increasingly designers are having to deal with the issue of people engaged in multiple interactions with different devices in parallel. One important commentator, Bruce 'Tog' Tognazinni, prefers the term 'interaction architect' to describe this profession.

How we got here

The revolution that has brought us to where we are today started towards the end of the Second World War, in 1945, with the development of the first digital computers. These were huge machines housed in specially built, air-conditioned rooms. They were operated by scientists and specialist computer programmers and operators, who physically pressed switches and altered circuits so that the electronics could complete their calculations.

During the 1960s computer technology was still dominated by scientific and accounting applications. Data was stored on paper tape or cards with holes punched in them, on magnetic tapes and large magnetic disks, and there was little direct interaction with the computer. Cards were sent to the computer centre, data was processed and the results were returned a few days later. Under the guidance of 'Lick' Licklider, however, things were beginning to change. The first screens and cathode ray tubes (CRTs) were being used as interactive devices and the first vision of a computer network – an internet – was formulated by Licklider. He worked at the Advanced Research Projects Agency (ARPA) at the US Department of Defense. His work also led to the establishment of computer science at four US universities (Licklider, 2003). Licklider was followed by the pioneering work of Ivan Sutherland at MIT, Doug Englebart who is credited with inventing the computer mouse, and Ted Nelson who developed the concept of hypertext, the idea of linking objects and being able to jump directly from one object to the next. In the UK pioneering work on computers was based at Manchester University and in 1959 Brian Shackel had published the paper 'Ergonomics for a computer'.

During the 1970s computing technology spread into businesses and screens linked to a central computer began to emerge. Computers were becoming networked together and indeed the first e-mail was sent over the ARPANET in 1972. The method of interaction for most people in the 1970s was still primarily 'batch'; transactions were collected together and submitted as a batch of work and computing power was shared between different people. Interest in HCI began to grow, with publications in the *International Journal of Man–Machine Studies*. As the decade ended so keyboards and screens became more common, but it was not until 1982 that the first real graphically based interfaces appeared in the form of the Xerox Star, Apple Lisa and Apple Macintosh computers. These used a bit-mapped display, allowing a graphical user interface (GUI) and interaction through pointing at icons and with commands grouped into menus. This style became ubiquitous when, in 1985, the Windows operating system appeared on (what were then usually IBM) personal computers (PCs). The personal computer and Windows-like operating system are attributed to another important pioneer, Alan Kay. Kay obtained his PhD, studying under Ivan Sutherland, in 1969, before moving to Xerox Palo Alto Research Center (PARC). It was here that the object-oriented computer programming language Smalltalk was developed. Many argue that it was the development of the VisiCalc spreadsheet program on the Apple

→ Chapter 12 discusses GUIs

II computer (the 'killer app') in 1979 that really fired the personal computer market (Pew, 2003).

The 1980s was the decade of the microcomputer, with the BBC Micro home computer selling over 1 million units and a whole plethora of home computers being adopted worldwide. Games consoles were also gaining in popularity in the home entertainment market. In business, people were getting networked and the Internet began to grow, based around e-mail. It was during the 1980s that human–computer interaction (HCI) came of age as a subject. In both the USA and Europe the first big conferences on HCI were held: the CHI '83 conference on Human Factors in Computing Systems in Boston, MA, and INTERACT '84 in London. Don Norman published his famous paper 'The trouble with UNIX: the user interface is horrid' (Norman, 1981) and Ben Shneiderman published *Software Psychology* (Shneiderman, 1980).

In the 1990s colour and multimedia arrived on the PC, which had begun to dominate the computer market. In 1993 a new interface was produced that took advantage of a simple mark-up or specification 'language' (called hypertext mark-up language, HTML). Thus the 'World Wide Web' came about and revolutionized the whole process of transmitting and sharing files. Pictures, movies, music, text and even live video links were suddenly available to everyone at work and at home. The growth of personal, community and corporate websites was phenomenal and the vision of a wholly connected 'global village' community began to become a reality. Of course, this growth was primarily in the West and in the USA in particular, where 'broadband' communications enabled a much more satisfying experience of the Web than the slow connections in Europe. Many parts of the world were not connected, but in the twenty-first century connections to the Web are global.

By the turn of the century the convergence of communications and computing technologies was just about complete. Anything could potentially be connected to anything, anywhere. Since all the data was digital, it could all be transmitted over the airwaves or over wired networks, and it could easily be transformed from one form into another. The proliferation of mobile devices, coupled with the wide availability of the Internet, brings us to the age of 'ubiquitous computing', a term first coined by the late Mark Weiser in 1993 when he talked of interaction through 'pads, tabs and boards'. Computing devices are now pervasive amongst people and across the world, providing all manner of services and experiences. Computing power continues to double every 18 months or so (according to Moore's law), producing mobile devices that are more powerful now than the largest computers were even just a few years ago. In the twenty-first century computing is truly ubiquitous and interaction is increasingly through touch and gesture rather than the keyboard that has been the main method of input since the PC revolution began. We now have Weiser's pads, tabs and boards in the form of phones and tablets in various sizes, large public screens and wearable computing (Figure 1.8). They all have access to the Web and run different apps. A huge amount of data is stored, and there are billions of videos on YouTube and photos on Flickr. Everything is synchronized and stored in the 'cloud' (in reality the cloud is a network of vast data centres full of computers) and broadband, wireless connectivity is becoming increasingly fast. The interconnectivity provided by the Web and wireless communications makes this a fascinating time to be an interactive systems designer.

Where are we heading?

It is a brave person who makes any strong prediction about where new technologies are headed as there are so many confounding factors. It is never just a technology that wins, but technology linked with a good business model linked with timing. Don

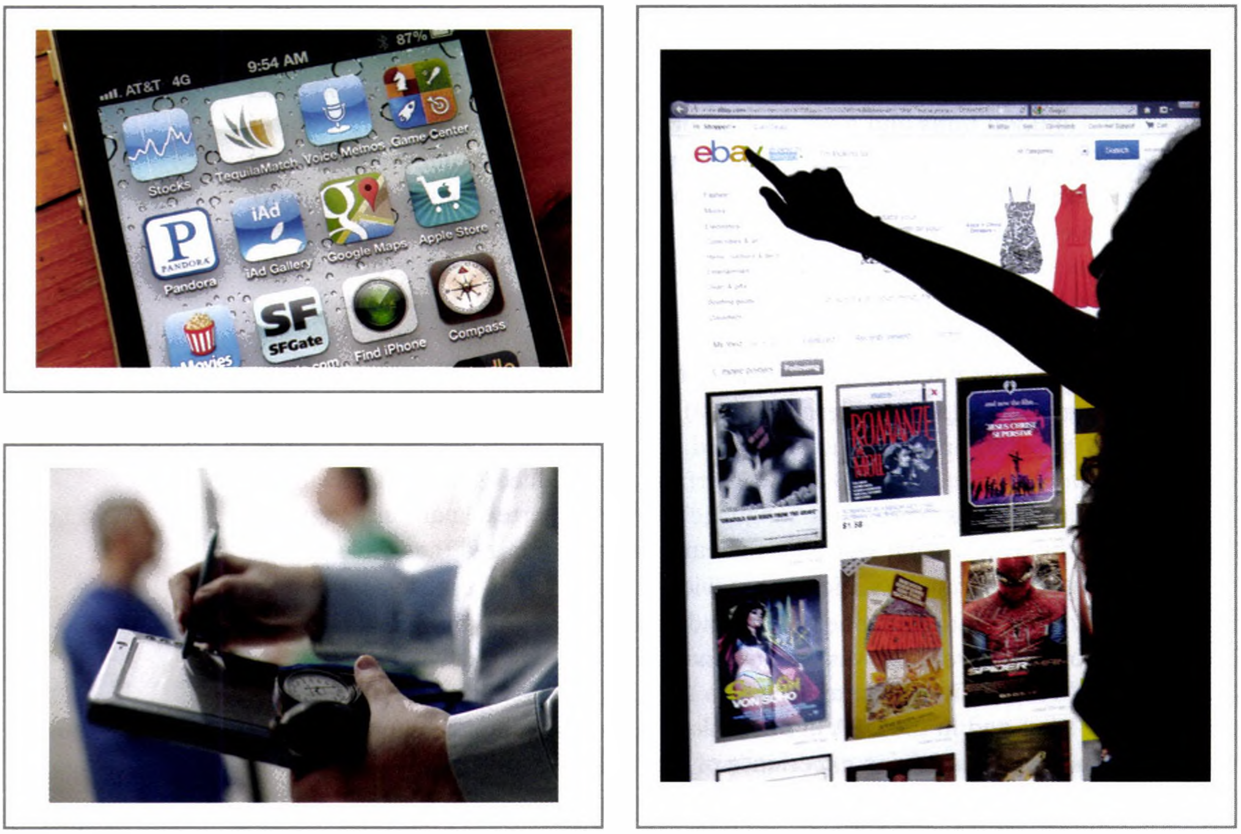

Figure 1.8 Tabs, pads and boards
(Source: (tl) Justin Sullivan/Getty images; (bl) Comstock/Alamy Images; (r) Bryan Bedder/Getty Images)

Norman delivers an interesting insight into both the past and future of technologies in his book *The Invisible Computer* (1999). Discussing such things as why the VHF video format succeeded over Betamax and why Edison's phonograph was not as successful as Emile Berliner's, he takes us forward to something he calls 'information appliances'. This notion has been taken up by others (Sharpe and Stenton, 2003), providing the following set of characteristics of information appliances:

- Appliances should be everyday things requiring only everyday skills to use.
- Appliances have a clear, focused function that can be used in a variety of circumstances.
- Peer-to-peer interaction. A key idea of appliances is that they work together without the need for central control or uploading and downloading.
- Direct user interface. Appliances need to be simple and intuitive to use.
- Successful appliances are those which support the notion of the swift and simple completion of a task.
- Appliances represent the ability to do something on impulse without having to think hard about how to do it.
- Appliances are personal and portable.

In 2013 this vision has been achieved to some extent with the range of smartphones such as the iPhone and Samsung Galaxy. But rather than the appliance concept being reflected in hardware, it is provided through the thousands of focused applications ('apps') that are available to download on to the iPhone, the Google Android or one of the other mobile platforms. Indeed Google along with Amazon are pioneering the idea of cloud computing where you don't need to carry any applications or data with you; just keep them in the cloud and download them when you need them.

FURTHER THOUGHTS

Whom do you trust?

Wireless connectivity between devices is now common both through the 'Wi-fi' standard called IEEE 802.11 and through Bluetooth. For example, your mobile phone will connect to your laptop computer via Bluetooth, and the laptop may be connected to an internal company network via a wireless network and hence to the Internet through the company's wired connection and hence to any other device in the world. How will you know where any piece of data that you look at actually is? If you look at the address book 'in your phone', you might in reality be accessing an address book on your laptop, or on any computer on the company's network or indeed anywhere on the World Wide Web. If data is duplicated, how will it be kept consistent? Across which devices will the consistency be reliable?

What we do know is that new products, business models, services and a range of other features will rapidly come into the world, and the interactive systems designer has to be ready to cope. Whether information appliances are just one of many directions that the future takes, we will have to see. In Microsoft's vision of HCI in 2020 (Microsoft, 2008) they argue that 'HCI needs to move forward from concerns about the production and processing of information toward the design and evaluation of systems that enable human values to be achieved' (p. 77) – something also emphasized by Cockton (2009) and his call for worth-centred design and Bødker in her consideration of 'third wave' HCI (Bødker, 2006)

The design company IDEO undertakes a wide range of projects in interactive systems design as illustrated through some of their projects illustrated in Figure 1.9 (the project shown in Figure 1.9 dates back to 2001). Some projects explore different ideas of changing concepts such as identity, others aim to produce new products and others look to see how people use technologies in their daily lives.

Figure 1.9 Concepts for future business cards and ideas of identity
(Source: IDEO, 2003. Courtesy of IDEO)

Challenge 1.3

Visit the website of IDEO and look at their projects. Talk about the ideas with a friend.

1.4 The skills of the interactive systems designer

Designers of interactive systems need a variety of skills and need to understand a variety of disciplines if they are to be able to do their jobs well. They need the mixture of skills that allows them to be able to:

- Study and understand the activities and aspirations of people and the contexts within which some technology might prove useful and hence generate requirements for technologies
- Know the possibilities offered by technologies
- Research and design technological solutions that fit in with people, the activities they want to undertake and the contexts in which those activities occur
- Evaluate alternative designs and iterate (do more research and more design) until a solution is arrived at.

The range of skills and academic disciplines that will contribute to such a person is significant. Indeed, it is often the case that no single person possesses all the skills needed for some design activity, which is why the design of interactive systems is often an affair for a design team. An interactive systems designer may be involved in a community information system project on one occasion, a kiosk for processing photographs on another, a database to support a firm of estate agents on another, and a children's educational game on another! Designers of interactive systems cannot be expert in all these fields, of course, but they must be aware enough to be able to take techniques from different areas, or access research in different disciplines when appropriate. We group the subjects that contribute to the design of interactive systems under the headings of knowledge of people, technologies, activities and contexts, and design, and illustrate the relationships in Figure 1.10 (p. 20).

People

→ Chapter 7 includes a discussion of ethnography

→ Chapter 23 discusses cognitive psychology and embodied cognition

People are social beings, so it is important that the approaches and techniques adopted in the social sciences are used to understand people and technologies. Sociology is the study of the relationships between people in society, the social, political and other groups that they participate in, and the settings in which such relationships take place. Anthropology is similar but focuses also on the study of culture, biology and language and on how these have evolved and changed over time. Both use techniques such as interviews and observation to arrive at their conclusions. A key approach, particularly in anthropology, is 'ethnography', which uses qualitative methods such as observations and unstructured interviews to produce a description of a particular culture or social group and its setting. Also related is cultural studies, which looks at people and their relationship with cultural issues such as identity, but also much more prosaic cultural activities such as shopping, playing computer games or watching TV. Descriptions tend to be from a more literary criticism background, informed by experience and reflection. Psychology is the study of how people think, feel and act. In particular, cognitive psychology seeks to understand and describe how the brain functions, how language works and how we solve problems. Ergonomics is the study of the fit between people and machines. In designing interactive systems, the designer will borrow much from each of these disciplines, including methods to help understand and design for people.

Technologies

The technologies that interactive systems designers need to know about include both software and hardware. Software engineering has developed methods for specifying and implementing computer programs. Programming languages are used to issue instructions to any programmable device such as a phone, computer, robot dog or earrings, shirts and chairs. Designers need to be aware of hardware for sensing different types of data (sensors) and for bringing about some change (actuators, or effectors). There are many different components available that produce many different effects and here designers will draw upon engineering knowledge, principles and methods. Communication between devices uses various communication 'protocols'. Designers need to know how different devices can communicate.

Activities and contexts

Interaction will usually take place in the context of some 'community of practice'. This term is used to denote groups of people who have shared interests and values and engage in similar activities. In business communities and organizations, information systems methods have developed over the years to ensure that information systems are developed that are effective and meet the needs of people who work there. In particular, soft systems theory (Checkland and Scholes, 1999) provides a very useful framework for focusing on the design of interactive systems. Social and organizational psychology are needed to look at the effects of technological change on organizations, and recently knowledge management and social computing have become important areas. Finally, new technologies offer new opportunities as business and interactive systems designers find that they are sometimes creating whole new ways of working with their designs.

Design

Principles and practices of design from all manner of design disciplines are used in designing interactive systems. Ideas and philosophy from architecture, garden design, interior design, fashion and jewellery design all crop up in various ways and different forms. It is not easy to simply pick up ideas from design disciplines, as much design knowledge is specific to a genre. Designers need to know the materials they work with and it is likely that more specialist design disciplines will emerge. One such discipline is product design, which is itself changing as it takes on board the nature of interactivity. Product design is an important contributing discipline to the skills of the designer of interactive systems. Graphic design and information design are particularly important for issues of information layout and the understandability and aesthetic experience of products. Human–computer interaction has itself evolved many techniques to ensure that designs are people-focused.

→ Chapter 12 discusses information design

Challenge 1.4

Imagine that you are put in charge of a design team that is to work on a project investigating the possibility of a new set of Web services for a large supermarket. These services will allow connection from any fixed or mobile device from any location, allowing food items to be ordered and delivered. The client even wants to investigate the idea of a 'smart refrigerator' that could automatically order items when it ran out. What range of skills might you need and which subject areas would you expect to draw upon?

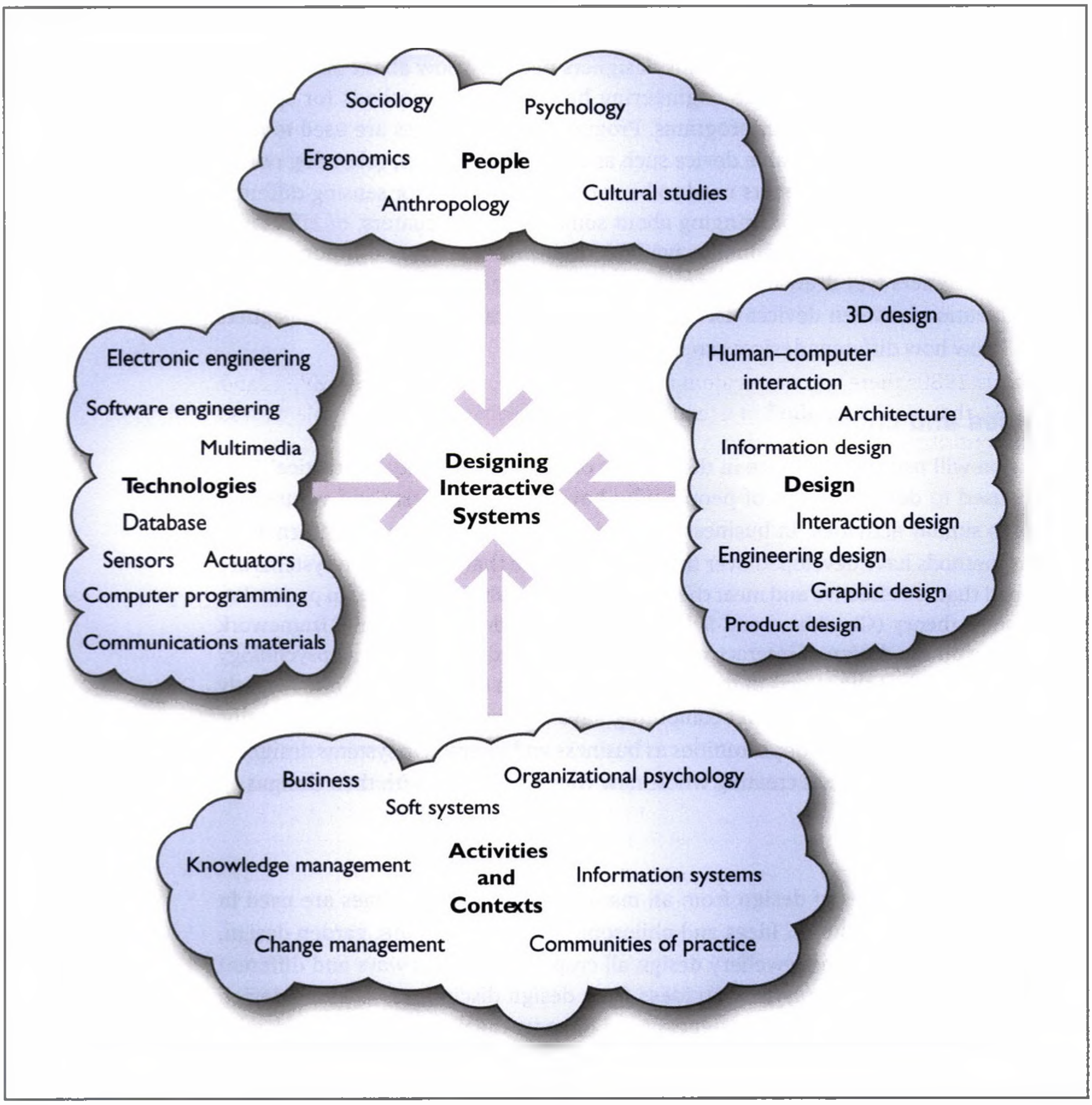

Figure 1.10 Disciplines contributing to interactive systems design

1.5 Why being human-centred is important

Being human-centred in design is expensive. It involves observing people, talking to people and trying ideas out with people, and all this takes time. Being human-centred is an additional cost to any project, so businesses rightly ask whether taking so much time to talk to people, produce prototype designs and so on is worthwhile. The answer is a fundamental 'yes'. Taking a human-centred approach to the design of interactive systems is advantageous for a number of reasons.

Return on investment

Williams *et al.* (2007) provide details of a number of case studies looking at the costs of taking a human-centred approach to interactive systems design and at the benefits that arise. Paying attention to the needs of people, to the usability of the product, results in reduced calls to customer helplines, fewer training materials, increased throughput, increased sales and so on.

Involving people closely in the design of their systems will help to ensure acceptability. Systems will be more effective if they are designed from a human-centred perspective and people will be more productive. Nowhere is the economic argument more pertinent than in Web design and e-commerce sites. Jared Spool and his company User Interface Engineering have a number of reports demonstrating the importance of good design to e-commerce and claim that sales can be increased by 225 per cent by turning 'browsers' into 'buyers'.

Safety

In the early 1980s there was an accident at a nuclear power plant at Three Mile Island in the USA that almost resulted in a 'meltdown'. Reportedly one of the problems was that the control panel indicated that a valve was closed when it was in fact open, and another indicator was obscured by a tag attached to another control: two fundamental design errors – one technical and one organizational – that human-centred design techniques would help to avoid. Other classic horror tales include a number of plane and train disasters that have been attributed to faulty displays or to operators not understanding or interpreting displays correctly. Systems have to be designed for people and for contexts. It is no good claiming 'human error' if the design was so bad in the first place that an accident was waiting to happen.

Ethics

Being human-centred also ensures that designers are truthful and open in their design practice. Now that it is so easy to collect data surreptitiously and to use that data for purposes other than what it was intended for, designers need to be ever more vigilant. As systems are increasingly able to connect autonomously with one another and share data it is vital that people know where the data that they give is going and how it might be used. People need to trust systems and be in a position to make choices about privacy and how they are represented.

The issue of intellectual property is another important aspect of ethical design; it is very easy to take an image from a website and use it without giving proper acknowledgement for its source. There are many issues associated with plagiarism or other dishonest uses of written materials. Privacy, security, control and honesty are all significant features of the interactive systems designer's life. Equality and attention to access are two of the 'political' issues that designers must address.

As technology changes so do traditional views and approaches to big moral and ethical questions. There are standards and legal requirements that need to be met by designs. Fundamentally, ethical design is needed because the systems that are produced should be easy and enjoyable to use, as they affect the quality of people's lives. Designers have power over other people and must exercise that power in an ethical fashion. The ACM (Association of Computing Machinery) code of ethics gives good advice on ethical design.

Sustainability

Interactive systems have a big impact on the world, and designers should approach interaction design from the perspective of what is sustainable. Millions of mobile phones and other devices are thrown away each year and they contain metals that are potentially dangerous to the environment. Large displays and projectors gobble up power. Cultures get swamped by the views and values of the dominant suppliers of hardware and software and local languages die out when all information is in English, Chinese or Hindi. Human-centred design needs to recognize diversity and design to enhance human values.

Summary and key points

Designing interactive systems is a challenging and fascinating discipline because it draws upon and affects so many features of people's lives. There is a huge variety of interactive systems and products, from business applications of computers to websites to dedicated information appliances to whole information spaces. Designing interactive systems is concerned with designing for people using technologies to undertake activities in contexts. Designing interactive systems needs to be human-centred.

- It draws upon many different subject areas, including both engineering design and artistic design.
- It is needed because we live in a digital age when bits are easily transformed and transmitted.
- It is necessary if we are to have safe, effective, ethical and sustainable design.

Exercises

1 Spend some time browsing the websites of corporations such as IDEO, Sony and Apple. Do not just look at the design of the site (though that can be useful); look at the products they are talking about and the philosophy of their design approach. Collect together your favourites and be prepared to spend time discussing them with your colleagues. Think of the whole range of issues about the site: what it looks like, how easy it is to use, how relevant the content of the site is, how clearly the content is organized, what the overall 'feel' of the site is.

2 Being human-centred is about
   - Thinking about what people want to do rather than what the technology can do
   - Designing new ways to connect people with people
   - Involving people in the design process
   - Designing for diversity.

   Write down how you might approach the design of the supermarket shopping service discussed in Challenge 1.4. Don't do the design; think about how to approach the design. Are there any issues of effectiveness, safety, ethics and sustainability that need to be considered?

Further reading

Laurel, B. (ed.) (1990) *The Art of Human–Computer Interface Design.* Addison-Wesley, Reading, MA. *Although this book is quite old, many of the articles in it are still relevant and many of the authors of those articles are still at the forefront of interaction design today.*

Norman, D. (1999) *The Invisible Computer: Why Good Products Can Fail.* MIT Press, Cambridge, MA. *This is an enjoyable book to read about successes and failures, pasts and futures for technologies.*

Getting ahead

Friedman, B. and Kahn, P.H. (2007) Human values, ethics and design. In A. Sears and J. A. Jacko (eds) *The Human–Computer Interaction Handbook*, 2nd edn. Lawrence Erlbaum Associates, Mahwah, NJ.

Norman, D. (1993) *Things That Make Us Smart*. Addison-Wesley, Reading, MA.

Norman, D. (1998) *The Design of Everyday Things.* Addison-Wesley, Reading, MA. *These two easy-to-read books provide a wealth of examples of good and bad design.*

Web links

The Usability Professionals Association is at **www.upassoc.org**

The Interaction Design Association is at **www.ixda.org**

The on-line material that goes with this chapter is at **www.pearsoned.co.uk/benyon**

Comments on challenges

Challenge 1.1

Of course, what you say will be dependent on the product or systems chosen. The important thing is to think in broad terms about the nature of the interaction with the device and at the activities that the device enables, and how good it is at doing them!

I could talk about the coffee machine at work, which is a simple, functional device. A single button press produces a reasonable cup of coffee. It is limited, however, in the variety of coffees that I can get (four types only) so I would ideally prefer a person mixing coffee for me rather than getting it from a machine. If I stay late at work and have to use the other coffee machine, it is a nightmare. The money slots don't work properly, the cups are too thin so the drink burns your hands, and the default is coffee with sugar (which I hate) so I have to remember to press the 'no sugar' button. Which I frequently forget to do!

This simple device can be contrasted with a website. Choose a website you like to visit. Discuss the opening page. Is it nice and clean, is there a site map, or other help to get the visitor oriented? How is the site using images and how does it work with the information? Are things difficult to read or difficult to control? Look at how users have to navigate from one page to another.

Challenge 1.2

The interface to the microwave consists of the various switches on the front that allow programming the time and temperature. There is also an audio part – the 'ping' when the timing is finished. The remote control just uses buttons as the interface and the Xbox controller has various buttons and a 4-way joystick. The PDA uses a pen (pointer) and a touch-sensitive screen. Icons are used on the screen and there are a few buttons on the casing. The PDA accepts 'graffiti' handwriting recognition.

Challenge 1.3

The aim of this challenge is to get you to think beyond user interfaces and beyond human–computer interaction to the changes that new technologies are bringing or could bring. As we create new information appliances and new products such as business cards, we, you, interactive systems designers, change the world. We change what is possible and change how people interact with other people. Reflect on (and discuss with someone else, if possible) the political, moral and ethical issues of these concepts.

Challenge 1.4

This project will demand a wide range of skills. On the technology side there are networking and software engineering issues concerned with how devices can be programmed to do this and how the information about products and orders can be stored. There will be issues of authorization and authentication of payments. Product design may come in if there are to be purpose-built devices created to access the services (e.g. an in-store smart scanner that could be used to record items bought). There will be a lot of information design expertise required and some graphic design to help in the layout of information. On the people side of things, general psychological knowledge will help to inform the design, and sociology may help to understand the social setting and impact that such services would have. Business models may need to be developed and certainly the skills of information systems designers will be needed.

Chapter 2

PACT: a framework for designing interactive systems

Contents

Aims

An essential part of our approach to designing interactive systems is that it should put people first: it should be human-centred. We use the acronym PACT (people, activities, contexts, technologies) as a useful framework for thinking about a design situation. Designers need to understand the people who will use their systems and products. They need to understand the activities that people want to undertake and the contexts in which those activities take place. Designers also need to know about the features of interactive technologies and how to approach designing interactive systems.

After studying this chapter you should be able to:

- Understand the relationship between activities and technologies
- Understand the PACT framework
- Understand the main characteristics of people that are relevant to designing interactive systems
- Understand the main issues of activities and the contexts in which they occur
- Understand the key features of interactive technologies.

2.1 Introduction

People use technologies to undertake activities in contexts. For example, teenagers use mobile (cell) phones to send text messages to their friends while sitting on a bus. Secretaries use Microsoft Word to write documents in a firm of solicitors. Air traffic controllers work together to ensure the smooth operation of an airport. A septuagenarian woman presses various buttons to set the intruder alarms in her house. People use Facebook to make contact with other people when sitting in an Internet café.

In all these settings we see people using technologies to undertake activities in contexts and it is the variety of each of these elements that makes designing interactive systems such a difficult and fascinating challenge. Technologies are there to support a wide range of people undertaking various activities in different contexts. If the technology is changed then the nature of the activities will also change. This issue is nicely summed up in Figure 2.1.

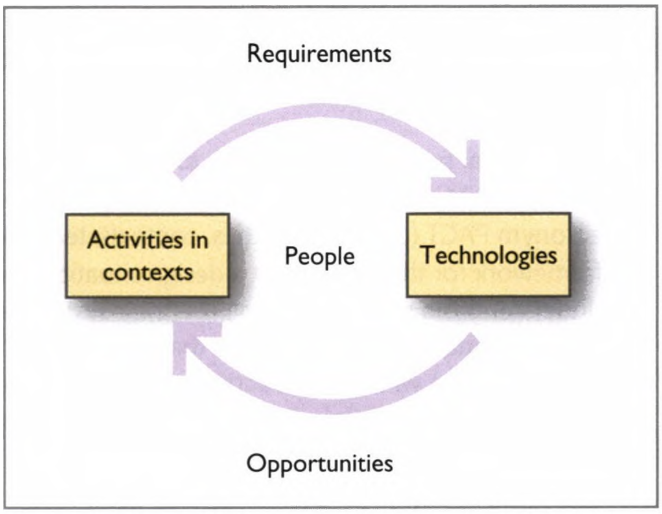

Figure 2.1 Activities and technologies

(Source: Based on Carroll (2002), Figure 3.1, p. 68)

Figure 2.1 shows how activities (and the contexts within which they take place) establish requirements for technologies that in turn offer opportunities that change the nature of activities. And so the cycle continues as the changed activity results in new requirements for technologies and so on. Designers need to keep this cycle in mind as they attempt to understand and design for some domain. (The word 'domain' here means an area of study, a 'sphere of activity'.) For example, as personal computers have become more common so the domain of e-mail has changed. Originally e-mail was all in text only, but now it is in full colour with pictures and video embedded. Other items can be attached to e-mails easily. This has led to a need for better facilities for managing it for organizing pictures, documents and addresses. Software now keeps track of threads of e-mails and links between e-mails. Another example is illustrated in Figure 2.2.

Challenge 2.1

Think of the activity of watching a film. List some ways in which this activity has changed with the introduction of video cassette recorders (VCRs) and digital versatile discs (DVDs) and downloads onto a laptop. How have the contexts changed since the early days of cinema?

To design interactive technologies we need to understand the variety inherent in the four elements of PACT.

Figure 2.2 The changing nature of telephoning activity as technology advances
(Sources: Press Association Images; Susanna Price/DK Images; Mike van der Wolk/Pearson Education Ltd.)

2.2 People

There can be few less controversial observations than that people differ from one another in a variety of ways. The chapters in Part IV of this book deal with these differences in detail. Here we summarize some of the most important features.

Physical differences

People differ in physical characteristics such as height and weight. Variability in the five senses – sight, hearing, touch, smell and taste – has a huge effect on how accessible, how usable and how enjoyable using a technology will be for people in different contexts. For example, colour blindness (usually the inability to distinguish correctly between red and green colours) affects about 8 per cent of Western males, short-sightedness and long-sightedness affect many, and many people are hearing-impaired. In Europe there are 2.8 million wheelchair users, so designers must consider where technologies are placed; and many people have dexterity impairments involving the use of their fingers. All of us have relatively large fingers compared to the small size we can use for buttons. Look at the ticket machine in Figure 2.3. What are the physical aspects of people that need to be taken into account in the design?

Ergonomics

The term 'ergonomics' was coined in 1948 to describe the study of the relationships between people and their environment. At that time, technically advanced weapons systems were being rapidly developed, which required that their design matched human and environmental factors if they were to be used effectively and, paradoxically, safely.

The environment includes the ambient environment (temperature, humidity, atmospheric pressure, light levels, noise and so on) and the working environment too (the design of machines, health and safety issues – e.g. hygiene, toxicology, exposure to ionizing radiation, microwaves, etc.).

Ergonomics is multidisciplinary, drawing on anatomy and physiology, various aspects of psychology (e.g. physiological and experimental), physics, engineering and work studies among others. In everyday life we come across the application of ergonomic design principles in every well-designed interactive system. In the advertisement for a new motor car, we can expect to find reference to its ergonomically designed dashboard

Figure 2.3 Metro ticket machine
(Source: Jules Selmes/Pearson Education)

(a good, desirable feature) or an adjustable, ergonomic driving seat. In the Mercedes-Benz sales literature for its new coupé we find the following ergonomic description:

> Once inside the C-Class Sports Coupé you'll find a wealth of ergonomic detail, designed to live up to the promise of its looks. As if cast from a single mould, the dashboard curves are smooth to the touch.

The term 'ergonomic design' is also extensively used of all manner of office furniture (chairs, desks, lights, footrests and so forth) and office equipment, for example keyboards, monitor stands and wrist rests. Many, if not most, of these principles are now embodied in legally binding design guidelines (see Further reading at the end of this chapter). Figure 2.4 is an illustration of an ergonomically designed keyboard. It is described as ergonomically designed as it reflects the fact that we have two hands – hence the two separate blocks of keys and an integral wrist support. The keyboard has been designed to match the hands and fingers of its intended users.

BOX 2.1

Anthropometrics

Anthropometrics means literally the measurement of man. Anthropometrics can, for example, tell us the limits (diameter and load-bearing characteristics) of the human wrist for the average man and woman. Figures have been compiled from thousands of measurements of different races, different ages and different professions (e.g. office workers vs manual workers) and drawn up as tables. The same body of data will also tell the designer whether the average person can simultaneously press button A while holding down buttons B and C – and whether this is true for both right- and left-handed people.

BOX 2.2

The changing role of the thumb

People who have grown up with mobile phones (or Game Boys) tend to use their thumbs when others are more likely to use their fingers. Sadie Plant from Warwick

Figure 2.4 An ergonomic keyboard

(Source: Microsoft Natural Multimedia Keyboard from www/microsoft.com/press/gallery/hardware/NaturalMultiMedia Keyboard.jpg © 2004 66Microsoft Corporation. All rights reserved. Printed with permission from Microsoft Corporation)

University (*New Scientist*, No. 2315, 3 November 2001) collected data on mobile phone usage in nine cities around the world, including Beijing, Chicago, London and Tokyo. She found that the under-25 age group appear to have experimented with the best way to interact with a mobile phone, one result of which is that now they use their thumbs to ring doorbells, push doors and point.

While ergonomics has a longer history than HCI, it would be a mistake to perceive it as being old and out of touch – quite the reverse. Ergonomics has much to tell us about the design of interactive devices such as a mobile games console, a tablet PC or smartphone. Figure 2.5 shows an example of the former.

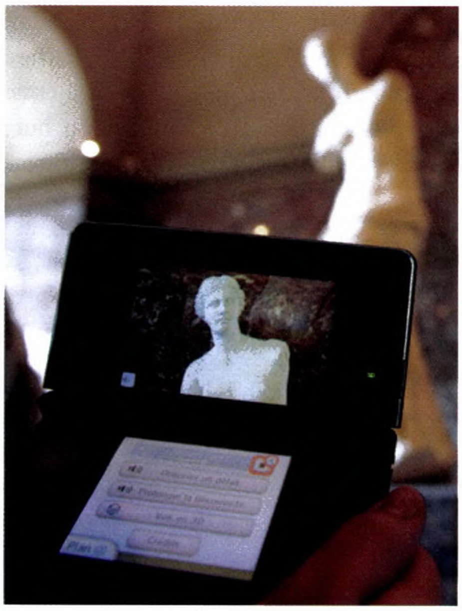

Figure 2.5 HP iPAQ Pocket

(Source: Patrick File/Getty Images/AFP)

Such devices are faced with ergonomic design challenges. For example, we all have relatively fat fingers compared with how small buttons can be made. In the world of mobile computing, small is good but too small is bad (too easily lost, too difficult to use, too easily eaten by the dog). Ergonomics can put numbers on what constitutes small and usable and what is too small and unusable. The best-known example of ergonomic knowledge being applied to HCI issues is Fitts's law (see Box 2.3).

BOX 2.3

Fitts's law

Fitts's law is a mathematical formula which relates the time required to move to a target as a function of the distance to the target and the size of the target itself, say moving a pointer using a mouse to a particular button. It is expressed mathematically as follows:

$$T_{(\text{time to move})} = k \log_2(D/S + 0.5)$$

where $k \sim 100$ ms, D is the distance between the current (cursor) position and the target, and S is the size of the target.

Thus one can calculate the time to move a distance of 15 cm to a button of size 2 cm as

$$T = 100 \log_2\left(\frac{15}{2} + 0.5\right)$$
$$= 0.207 \text{ seconds}$$

Fitts's law describes motor control. The smaller the target and the greater the distance, the longer it will take to hit the target. Fitts's law can also be used to calculate how long it would take to type this sentence or, more importantly, a number of time-critical operations such as hitting the brake pedal of a motor car, or the likelihood of hitting <OK> rather than <Cancel> or, more worryingly, <Fire> or <Detonate>.

Psychological differences

Psychologically, people differ in a variety of ways. For example, people with good spatial ability will find it much easier to find their way around and remember a website than those with poor ability. Designers should design for people with poor ability by providing good signage and clear directions. Language differences are of course crucial to understanding, and cultural differences affect how people interpret things. For example, in the Microsoft Excel spreadsheet application there are two buttons, one labelled with a cross and the other a tick. In the USA a tick is used for acceptance and a cross for rejection, but in Britain either a tick or a cross can be used to show acceptance (e.g. a cross on a voting paper).

BOX 2.4

Individual differences

There are often large differences in the psychological abilities of people. Some people have a good memory, others less so. Some people can find their way around environments better than others, or mentally rotate objects more quickly and accurately. Some are good at words, others are good at numbers. There are differences in personality, emotional make-up and ability to work under stress. Many tests have been designed to measure these differences. For example the Myers–Briggs Type Indicator is a series of tests that results in people being classified as one of 16 personality types. Others classify people as one of five personality types known as OCEAN: Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. Designers need to consider the range of differences between people and the demands that their designs make on people's psychological abilities.

People also have different needs and abilities when it comes to attention and memory and these can change depending on factors such as stress and tiredness. Most people cannot remember long numbers or complicated instructions. All people are better at recognizing things than they are at remembering things. Some people can quickly grasp how something works, whereas for others it can take much longer. People have had different experiences and so will have different conceptual 'models' of things.

Mental models

The understanding and knowledge that we possess of something is often referred to as a 'mental model' (e.g. Norman, 1998). If people do not have a good mental model of something they can only perform actions by rote. If something goes wrong they will not know why and will not be able to recover. This is often the case with people using software systems, but it is also the case with 'simpler' domestic systems such as central heating systems, thermostats and so on. A key design principle is to design things so that people will form correct and useful mental models of how they work and what they do.

People develop mental models through interacting with systems, observing the relationship between their actions and the behaviours of the system and reading any manuals or other forms of explanation that come with a system. So, it is important that designers provide sufficient information in the interface (and any accompanying documentation) for people to form an accurate mental model.

Figure 2.6 illustrates the problem. As Norman set out in his classic exposition of the issues (Norman, 1986), designers have some conception of the system they have produced. This may or may not be the same as what the system actually does. Moreover, in a system of any large size, no single designer will know everything that the system does.

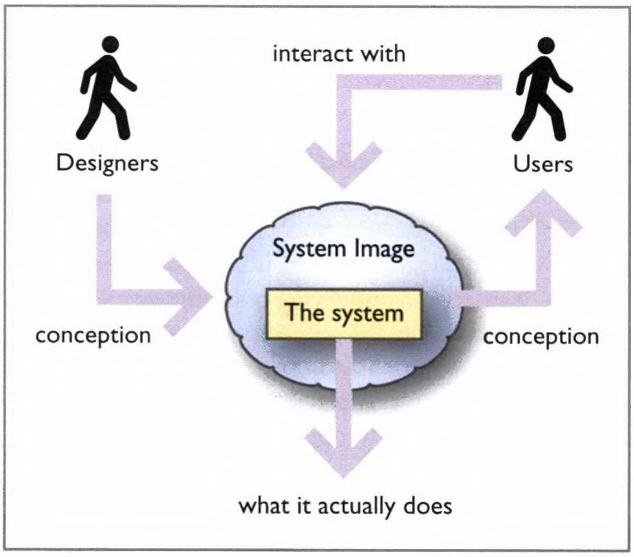

Figure 2.6 The system image

Designers design a system's image that they hope will reveal the designers' conception. The problem is that it is only through the system image – the interface, the behaviours of the system and any documentation – that the designers' conception can be revealed. People interact with the system image and from this have to derive their conception (their 'mental model') of what the system is and what it does. A clear, logical and consistent conceptual design will be easier to communicate to people who use the system and hence they will develop a clearer conception of the system themselves.

Norman has made the following general observations about the nature of mental models of interactive systems (Norman, 1983). He concludes that:

- Mental models are incomplete. People will understand some parts of a system better than others.
- People can 'run' (or try out) their models when required, but often with limited accuracy.
- Mental models are unstable – people forget details.
- Mental models do not have firm boundaries: similar devices and operations get confused with one another.
- Mental models are unscientific, exhibiting 'superstitious' behaviour.
- Mental models are parsimonious. People are willing to undertake additional physical operations to minimize mental effort, e.g. people will switch off the device and start again rather than trying to recover from an error.

→ Chapter 22 discusses the 'mind's eye' in terms of the visuospatial sketchpad of working memory

The psychologist Stephen Payne (1991, pp. 4–6) describes how mental models predict behaviour. The claim is that, in many situations, a great deal of explanatory work can be done by a description of what people know and believe, and how this affects their behaviour. Inferences can be made by 'mental simulation'. Mental models can support reasoning about devices, or the physical world in general, by running simulations in the mind's eye.

?

Challenge 2.2

What is your mental model of e-mail? How does an e-mail message get from one place to another? Write down your understanding and discuss it with a colleague. What differences are there and why? Think about the level of detail (or level of abstraction) that is present in different models.

BOX 2.5

Device models

Kieras and Bovair (1984) investigated the role of a device model (a person's mental model of a device) in learning how to operate a mock-up of the weapons control panel of the *USS Enterprise* from *Star Trek*. In their first experiment subjects learned how to operate the 'phasers' either by means of rote learning (press this button, then turn that knob to the second position) or by learning the underlying principles (the energy booster takes power from the ship), which required the subjects to infer the procedures. Kieras and Bovair found that learning, retention and use of 'shortcuts' were all enhanced for the group that learned the principles, demonstrating that knowledge of how the system worked enables people to infer how to operate it. Kieras and Bovair concluded by making two key points: first, for a device model to be useful it must support inference about exact and specific control actions, and secondly, the model need not be very complete or thorough.

Social differences

People make use of systems, products and services for very different reasons. They have different goals in using systems. They have different motivations for using systems. Some people will be very interested in a particular system, others will just want to get a simple task completed. These motivations change at different times.

Novice and expert users of a technology will typically have very different levels of knowledge and hence requirements for design features. Experts use a system regularly and learn all sorts of details, whereas a beginner will need to be guided through an interaction. There are also people who do not have to use a system, but who the designer would like to use the system. These people (sometimes called 'discretionary users') are often quickly put off if things are difficult to do. Designers need to entice these people to use their systems.

Designing for homogeneous groups of people – groups who are broadly similar and want to do much the same things – is quite different from designing for heterogeneous groups. Websites have to cater for heterogeneous groups and have particular design concerns as a result. A company's intranet, however, can be designed to meet the particular needs of particular people. Representatives from a relatively homogeneous group – secretaries or managers or laboratory scientists, say – could be made part of the design team and so provide much more detailed input as to their particular requirements.

Challenge 2.3

Look again at the ticket machine in Figure 2.3 and consider the people who will use it. Identify the variety of characteristics, physically, psychologically (including different mental models people might have) and socially, in terms of usage of the system.

2.3 Activities

There are many characteristics of activities that designers need to consider. The term is used for very simple tasks as well as highly complex, lengthy activities, so designers need to be careful when considering the characteristics of activities. Below is our list of the 10 important characteristics of activities that designers need to consider. First and foremost, the designer should focus on the overall *purpose* of the activity. After that the main features are:

- Temporal aspects (items 1–4)
- Cooperation (5)
- Complexity (6)
- Safety-critical (7 and 8)
- The nature of the content (9 and 10).

1 Temporal aspects cover how regular or infrequent activities are. Something that is undertaken every day can have a very different design from something that happens only once a year. People will soon learn how to make calls using a mobile phone, but may have great difficulties when it comes to changing the battery. Designers should ensure that frequent tasks are easy to do, but they also need to ensure that infrequent tasks are easy to learn (or remember) how to do.

2 Other important features of activities include time pressures, peaks and troughs of working. A design that works well when things are quiet can be awful when things are busy.

3 Some activities will take place as a single, continuous set of actions whereas others are more likely to be interrupted. If people are interrupted when undertaking some activity, the design needs to ensure that they can 'find their place' again and pick up. It is important then to ensure that people do not make mistakes or leave important steps out of some activity.

4 The response time needed from the system must be considered. If a website takes two minutes to deliver a response when the server is busy, that may be frustrating for a normal query but it could be critical if the information is needed for some emergency. As a general rule people expect a response time of about 100 milliseconds for hand–eye coordination activities and one second for a cause–effect relationship such as clicking a button and something happening. Anything more than five seconds and they will feel frustrated and confused (Dix, 2012).

→ There are many examples of cooperative activities in Chapter 18

5 Another important feature of activities is whether they can be carried out alone or whether they are essentially concerned with working with others. Issues of awareness of others and communication and coordination then become important.

6 Well-defined tasks need different designs from more vague tasks. If a task or activity is well defined it can be accomplished with a simple step-by-step design. A vague activity means that people have to be able to browse around, see different types of information, move from one thing to another and so on.

7 Some activities are 'safety-critical', in which case any mistake could result in an injury or a serious accident. Others are less so. Clearly, where safety is involved designers must pay every attention to ensuring that mistakes do not have a serious effect.

8 In general, it is vital for designers to think about what happens when people make mistakes and errors and to design for such circumstances.

9 It is also important to consider the data requirements of the activity. If large amounts of alphabetic data have to be input as part of the activity (recording names and addresses, perhaps, or word-processing documents) then a keyboard is almost certainly needed. In other activities there may be a need to display video or high-quality colour graphic displays. Some activities, however, require very modest amounts of data, or data that does not change frequently, and can make use of other technologies. A library, for example, just needs to scan in a barcode or two, so the technology can be designed to exploit this feature of the activity.

10 Just as important as data is the media that an activity requires. A simple two-tone display of numeric data demands a very different design from a full-motion multimedia display.

Challenge 2.4

List the main characteristics of the activity of sending an e-mail. Use the 10 points above to guide you.

2.4 Contexts

Activities always happen in a context, so there is a need to analyse the two together. Three useful types of context are distinguishable: the organizational context, the social context and the physical circumstances under which the activity takes place. Context can be a difficult term. Sometimes it is useful to see context as surrounding an activity. At other times it can be seen as the features that glue some activities together into a coherent whole.

For an activity such as 'withdraw cash from an ATM', for example, an analysis of context would include things such as the location of the device (often as a 'hole-in-the-wall'), the effect of sunshine on the readability of the display, and security considerations. Social considerations would include the time spent on a transaction or the need to queue. The organizational context for this activity would take into consideration the impact on the bank's ways of working and its relationships with its customers. It is important to consider the range of contexts and environments in which activities can take place.

Physical environment

The physical environment in which an activity happens is important. For example, the sun shining on an ATM display may make it unreadable. The environment may be very noisy, cold, wet or dirty. The same activity – for example, logging on to a website – may be carried out in geographically remote environments where Internet access is slow, or with all the facilities of a large city and fast networks.

Social context

The social context within which the activity takes place is also important. A supportive environment will offer plenty of help for the activity. There may be training manuals available, tuition or experts to hand if people get into trouble. There may be privacy issues to consider, and an interaction can be very different if the person is alone compared to being with others. Social norms may dictate the acceptability of certain designs. For example, the use of sound output is often unacceptable in an open-plan office environment, but might be quite effective where a person is working alone.

Organizational context

Finally the organizational context (Figure 2.7) is important as changes in technology often alter communication and power structures and may have effects on jobs such as deskilling. There are many books devoted to the study of organizations and the impact of new technologies on organizations. We cannot do justice to this subject here. The circumstances under which activities happen (time, place and so on) also vary widely and need to be taken into consideration.

Figure 2.7 Different working contexts

(Source: Peter Wilson/DK Images; Rob Reichenfield/DK Images; Eddie Lawrence/DK Images)

BOX 2.6

Interface plasticity

Joelle Coutaz and her colleagues (Coutaz and Calvary, 2012) present the idea of designing for interface plasticity. These are interfaces that adapt to different contexts, for example adapting a display of a heating controller from a display on the TV to a display on a small mobile device. Importantly they tie this in with the idea of designing for specific values. Designers should explicitly address the values that are being sought for people in a specific context. The interface should be designed to achieve the required values in the required contexts of use.

2.5 Technologies

The final part of the PACT framework is the technologies: the medium that interactive system designers work with. Interactive systems typically consist of hardware and software components that communicate with one another and transform some input data into some output data. Interactive systems can perform various functions and typically contain a good deal of data, or information content. People using such systems engage in interactions and physically devices have various degrees of style and aesthetics. Designers of interactive systems need to understand the materials they work with, just as designers in other areas of design such as interior design, jewellery design, etc. have to do.

Of course, interactive technologies change at a fantastic rate and by far the best way for designers to keep abreast of the options available is to subscribe to websites, a number of which are listed on the website that accompanies this chapter. It is also very difficult to classify technologies, as they are continually being packaged in new ways and different combinations facilitate quite different types of interactions. For example, the multi-touch screen on an iPod Touch allows for quite different ways of navigating through your music collection and selecting particular tracks from the trackwheel on an iPod Nano. Designers need to be aware of various possibilities for input, output, communication and content.

Input

Input devices are concerned with how people enter data and instructions into a system securely and safely. Switches and buttons facilitate a simple and direct method of issuing instructions (such as 'turn on' or 'turn off') but they take up space. On small mobile devices there is not enough room to have many buttons, so designers have to be careful which functions have their own button. On the iPhone, for example, a button on the side of the device is allocated to turning the sound off and on. The designers decided that this was such an important and often-used function that it should have its own button.

Alphanumeric data is usually input to an interactive device through a 'QWERTY' keyboard, invented by C.L. Sholes in 1868! At that time, typewriters were relatively crudely manufactured and an alphabetic arrangement of keys tended to result in jams when the keys were struck. By rearranging the keys Sholes solved this problem. The design is still with us today, despite some devices using an alphabetic keyboard where the letters are arranged in alphabetical order.

Touchscreens are sensitive to the touch of a finger. They function through either infrared sensitivity or electrical capacitance. Because of their lack of moving or detachable parts, they are suitable for applications intended for public places, and provided the interface is well designed they present an appearance of simplicity and ease of use. Many touchscreens only recognize a single touch, but multi-touch screens enable zooming and rotating of images and text. Figure 2.8 shows Microsoft's surface, a multi-touch table.

Touchscreens make use of the person's finger as the input device, which has the obvious benefit that people always have their fingers with them. The light pen (Figure 2.9) was, arguably, the original pointing device. When it is pointed at the screen it returns information about the screen location to a computer which allows the item pointed at to be identified. Light pens are less expensive than touchscreens, can be armoured (made very robust) and can be sterilized. They have a number of industrial and medical applications.

Other forms of pointing device include the stylus which is used on very small displays where a finger is too big to be used as the input device, and on many handheld devices. Being more precise than a finger, a stylus can be used for handwriting recognition. In theory, this is an attractive way of inputting data into an interactive device. Writing with a stylus directly onto a computer's screen or tablet is a natural way of working. However, it is quite slow and can be inaccurate. It requires people to 'train' the device to recognize their handwriting, which improves the recognition accuracy of the software. Many people can type faster than they can write by hand.

One of the most ubiquitous of input devices is the mouse (Figure 2.10), developed at Stanford University Research Laboratory in the mid-1960s. The mouse consists of a palm-sized device that is moved over a flat surface such as the top of a desk. At its simplest (and cheapest) it rests on a rubber-coated ball that turns two wheels set at right angles. These two wheels translate the movement of the mouse into signals that the computer to which it is connected can interpret. One or two buttons sit on top of the mouse and are operated with the person's fingers. The mouse has become the default pointing device. More contemporary mouse design includes a thumbwheel (see Figure 2.11) for scrolling through documents or Web pages. A mouse may be cordless, using infra-red to communicate with the host computer. In 2009 Apple introduced the 'magic mouse' that combined traditional mouse functions with multi-touch capability allowing a range of new touch gestures for interaction.

A trackball is another pointing device, which is best described as a mouse lying on its back. To move the pointer the user moves the ball. Again, like all other pointing devices, there are one or more buttons which can be used to select on-screen items. Trackballs are often found in public-access kiosks because they are difficult to steal and do not require a flat surface to rest upon.

Figure 2.8 Microsoft Surface
(Source: Reuters/Robert Sorbo)

Figure 2.9 A light pen
(Source: Volker Steger/Science Photo Library)

Figure 2.10 A Mac one-button mouse. The single button of the traditional Mac is said to have the advantage of 'you always know which button to press'

(Source: Alan Mather/Alamy Images)

Figure 2.11 A Microsoft two-button mouse with thumbwheel (which is used for scrolling)

(Source: www.microsoft.com/presspass/images/gallery/hardware/BNMS_mouse_web.jpg. Printed with permission from Microsoft Corporation)

A joystick (Figure 2.12) is a handle which pivots from a central point. Viewing the joystick from above, it may be moved north, south, east and west (and all points between) to control an on-screen pointer, spaceship or any other on-screen object. Joysticks are used mostly for computer games, but they are also found in conjunction with CAD/CAM (computer-aided design/manufacture) systems and VR (virtual reality) applications.

With the introduction of the Nintendo Wii in 2007 a whole new generation of input became possible. The Wii uses infra-red to register the movement of a wand. This allows gestures to be recognized. Other systems, notably the Microsoft Kinnect, recognize gestures through tracking limb and body movements by attaching sensors to the limb or by tracking using cameras (Figure 2.13).

There are many different types of sensor that are now available as input mechanisms. Air pressure sensors, acoustic sensors, vibration detectors, infra-red motion detectors and accelerometers are all readily available for designers to detect specific aspects of an

Figure 2.12 An ergonomically designed games joystick

(Source: Microsoft SideWinder ® Precision 2 joystick. Photo by Phil Turner. Printed with permission from Microsoft Corporation)

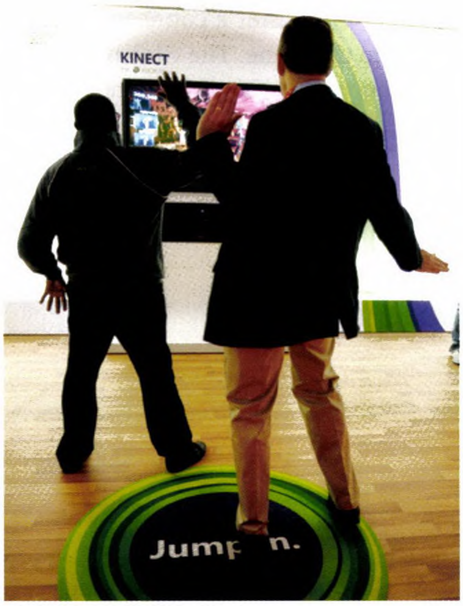

Figure 2.13 Microsoft Kinnect
(Source: David Becker/Getty Images)

interaction. Wilson (2012) lists sensors for detecting occupancy, movement and orientation, object distance and position, touch, gaze and gesture, human identity (biometrics), context and affect. There are many proprietary devices used to input specifically to mobile devices, such as jog wheels used for navigation of mobile phone interfaces. Brain activity can also be sensed, allowing for brain–computer interfaces (BCI) – an exciting development for the future.

Speech input is becoming increasingly accurate, particularly if people are willing to spend a few minutes (7–10, say) training a system to recognize their voice. Even without training, the Siri system on the iPhone can be quite impressive. The other day I said to Siri 'Send a text to Linda'. Siri replied 'There are two people called Linda in your address book, Linda and Linda Jane'. I replied 'Send a text to Linda Jane, say Hi'. Siri sent the text and responded, 'A text saying Hi has been sent to Linda Jane'. Speech input for these simple, focused tasks will become an increasingly common option for the interaction designer.

Other forms of input include quick response (QR) codes and augmented-reality (AR) fiducial markers (Figure 2.14). QR codes are used by a general-purpose scanning app on a phone to connect the phone to a website, or to execute a short sequence of operations.

Figure 2.14 QR codes
(Source: Red Huber/Orlando Sentinel/MCT/Getty Images)

Fiducial markers are used to recognize an object and hence to tailor some interactivity towards it. Markerless AR uses a photo of an object to register a connection allowing graphics, video and other content to be overlaid onto the scene. The Global Positioning System (GPS) can also be used to align views of the real world with digital content to provide an augmented reality.

Challenge 2.5

Which input devices would you use for a tourist information 'kiosk' application to be sited in the arrivals area of an airport? The system allows people to book hotel rooms, etc., as well as to find information about the area. Explain your choices.

Output

Technologies for displaying content to people rely primarily on the three perceptual abilities of vision, hearing and touch. The most fundamental output device is the screen or monitor. Even a few years ago the default monitor used cathode ray tube (CRT) technology that required a large heavy box positioned on a desk or table. Nowadays flat-screen monitors using plasma or TFT (thin-film transistor) or LCD (liquid crystal display) technologies can be mounted on walls. Some of these can deliver very large displays that result in a significantly different interactive experience. Flexible organic light-emitting diode (OLED) displays for screens are just coming onto the market that will enable displays of any shape and size that can bend and hence can be used in clothing (Figure 2.15)

→ Wearable computing is discussed at length in Chapter 20

The physical dimensions of display devices are, however, only one of the factors involved in the resulting output. The output device is driven by hardware – a graphics card – that will vary with respect to the screen resolutions and palette of colours it can support. More generally, designing interactive systems to work with any and all combinations of hardware is very difficult. Typically, applications and games specify minimum specifications.

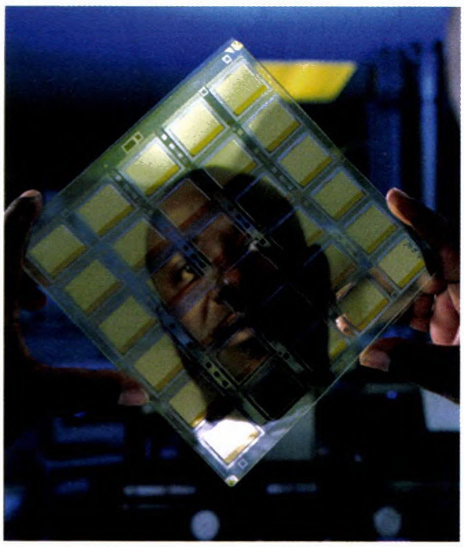

Figure 2.15 Flexible organic light-emitting diode (OLED) display

(Source: Volker Steger/Science Photo Library Ltd)

One way past the problems with restrictive display 'real estate' is to use a data projector (Figure 2.16). While the resolution is usually less than that of a monitor, the resulting projected image can be huge. Data projectors are shrinking in size at a remarkable rate and there are now mobile data projectors. These promise to have a big impact on interaction design as they get small enough to be built into phones and other mobile devices. Images can be projected onto any surface and pointing and other gestures can be recognized by a camera. In this way any surface has the potential to become a multi-touch display.

Besides the visual display of content, sound is an important method of output. Sound is an output medium that is significantly under-used. Speech output is also an increasingly popular option (e.g. in satellite navigation systems). With effective text-to-speech (TTS) systems, simply sending a text message to the system results in clear spoken output.

→ Sound is discussed at length in Chapter 13

A printer is a device that prints text or illustrations on paper, while a plotter draws pictures or shapes. Plotters differ from printers in that they draw lines using a pen. As a result, they can produce continuous lines, whereas printers can only simulate lines by printing a closely spaced series of dots. Multi-colour plotters use different-coloured pens. In general, plotters are considerably more expensive than printers.

Several companies have developed three-dimensional printers. These machines work by placing layers of a powdery material on top of each other to create a real-life model of a digital image. It is thought that with the use of hundreds and perhaps thousands of layers, everything from 'coffee cups to car parts' could be created. Like putting ink on paper, 3D printers print using powder and binder (glue). These printers allow for the rapid prototyping of physical designs for new products.

'Haptics' refers to the sense of touch. However, haptics allows us to be in touch with interactive devices and media in a way that is direct and immediate. Perhaps the most widespread haptic devices are those games controllers that incorporate so-called force-feedback. Force-feedback is typically intended to convey feedback from games environments back to the person engaged. So what are the perceived benefits of force-feedback devices?

→ Haptic interfaces are considered further in Chapter 13

- Sensations can be associated with interactions, such as feeling driving surfaces or feeling footsteps.
- Sensations can also be used to provide feedback as to the location of other players, objects and so forth.
- Force-feedback can allow the player to feel what it would be like to wield a sword, drive a high-speed car, fly a 'speeder' or engage the Empire with a light-sabre.

Figure 2.16 The Samsung i7410 Sirius Projector Phone
(Source: Gaustau Nacarino/Reuters)

A significantly more serious application of force-feedback is NASA's 'Softwalls' initiative in response to the 9/11 terrorist attacks on New York in 2001. Softwalls would be used to restrict airspaces by way of the aircraft's on-board systems. The basic idea, attributed to Edward Lee, would prevent aircraft from flying into restricted airspace (such as city centres) and this would be communicated to the pilot by way of the aircraft's joystick. Other examples include the 'silent alert' vibration of a mobile phone and even the feel of a key when pressed.

Challenge 2.6

Which output devices would you use for a tourist information application as described in Challenge 2.5? Explain your choices.

Communication

Communications between people and between devices is an important part of designing interactive systems. Here, issues such as bandwidth and speed are critical. So too is feedback to people so that they know what is going on and indeed that something is going on! In some domains the transmission and storage of large amounts of data becomes a key feature.

Communication can take place through wired connections such as a telephone line, or an Ethernet network which is often found in offices. Ethernet is the fastest form of communication, but the device has to be plugged into a network to make use of it. Ethernet allows connection to be made to the nearest node on the Internet. Extremely fast communications over fibre-optic cables connect these nodes to each other and hence connect devices to other devices all over the world. Each device on this network has a unique address, its IP (Internet Protocol) address, that enables data to be routed to the correct device. The number of IP addresses available will soon be used up and a new form of address, IPv6, will be needed.

Wireless communication is becoming much more common and often a wireless 'hub' is attached to an Ethernet network. Wireless communications can take place over the wireless telephone network used for mobile phones or over a Wi-Fi connection. Wi-Fi is quite limited in range and you need to be within a few metres of a Wi-Fi hub to get a connection, whereas over the telephone network, coverage is much wider. The new 4G technologies promise to deliver much faster connectivity over mobile devices and superfast broadband will soon be covering cities across the globe. Other forms of wireless communication continue to be developed and WiMax promises to deliver much wider coverage using Wi-Fi. Short-range communications directly between one device and another (i.e. not using the Internet) can be achieved using a technology called Bluetooth. Near-field communication (NFC) is used to connect devices simply by bringing them close to each other. All new mobile phones will soon have NFC capability, a feature which again will change the types of interaction that are possible.

Content

Content concerns the data in the system and the form it takes. Considerations of content are a key part of understanding the characteristics of the activities as described above. The content that a technology can support is also critical. Good content is accurate, up to date, relevant and well presented. There is little point in having a sophisticated information retrieval system if the information, once retrieved, is out of date or irrelevant. In some technologies content is just about everything (e.g. websites are usually all about content). Other technologies are more concerned with function (e.g. a remote control for a TV). Most technologies have a mixture of function and content.

Content can be retrieved when required (known as 'pull technology') or it can be pushed from a server to a device. Push e-mail, for example, is used on the BlackBerry system so that e-mail is constantly updated. RSS feeds on websites provide automatic updates when a website's content is changed.

The characteristics of the data are important for choosing input methods. Barcodes, for example, are only sensible if the data does not change often. Touchscreens are useful if there are only a few options to choose from. Speech input is possible if there is no noise or background interference, if there are only a few commands that need to be entered or if the domain is quite constrained.

'Streamy' outputs such as video, music and speech have different characteristics from 'chunky' media such as icons, text or still photographs. Most important, perhaps, is that streamy media do not stay around for long. Instructions given as speech output, for example, have to be remembered, whereas if displayed as a piece of text, they can be read over again. Animations are also popular ways of presenting content; 2D animation is generally produced using Adobe's Flash program and 3D-style animation can be produced with Papervision or games 'engines' such as 3D Studio Max and Maya.

2.6 Scoping a problem with PACT

The aim of human-centred interactive systems design is to arrive at the best combination of the PACT elements with respect to a particular domain. Designers want to get the right mix of technologies to support the activities being undertaken by people in different contexts. A PACT analysis is useful for both analysis and design activities: understanding the current situation, seeing where possible improvements can be made or envisioning future situations. To do a PACT analysis the designer simply scopes out the variety of Ps, As, Cs and Ts that are possible, or likely, in a domain. This can be done using brainstorming and other envisionment techniques and by working with people through observations, interviews and workshops. There are many techniques for this (these are described in Part II of this book). A PACT analysis is also useful for developing personas and scenarios (see Chapter 3). The designer should look for trade-offs between combinations of PACT and think about how these might affect design.

For people, designers need to think about the physical, psychological and social differences and how those differences change in different circumstances and over time. It is most important that designers consider all the various stakeholders in a project. For activities they need to think about the complexity of the activity (focused or vague, simple or difficult, few steps or many), the temporal features (frequency, peaks and troughs, continuous or interruptible), cooperative features and the nature of the data. For contexts they think about the physical, social and organizational setting, and for technologies they concentrate on input, output, communication and content.

As an example, let us assume that we have been asked by a university department to consider developing a system controlling access to their laboratories. A PACT analysis might include the following.

People

Students, lecturers and technicians are the main groups. These are all well educated and understand things such as swipe cards, passwords and so on. People in wheelchairs need to be considered, as do other design issues such as colour blindness. There may be language differences. Both occasional and frequent visitors need to be considered. However, there are other stakeholders who need access to rooms, such as cleaning staff and security personnel. What are the motivations for management wanting to control access in the first place?

Activities

The overall purpose of the activity is to enter some form of security clearance and to open the door. This is a very well-defined activity that takes place in one step. It happens very frequently, with peaks at the start of each laboratory session. The data to be entered is a simple numeric or alphanumeric code. It is an activity that does not require cooperation with others (though it may be done with others, of course). It is not safety-critical, though security is an important aspect.

Contexts

Physically the activity takes place indoors, but people might be carrying books and other things that makes doing anything complicated quite difficult. Socially it may happen in a crowd, but also it may happen late at night when no one else is about. Organizationally, the context is primarily about security and who has access to which rooms and when they can gain access. This is likely to be quite a politically charged setting.

Technologies

A small amount of data has to be entered quickly. It must be obvious how to do this in order to accommodate visitors and people unfamiliar with the system. It needs to be accessible by people in wheelchairs. The output from the technology needs to be clear: that the security data has been accepted or not and the door has to be opened if the process was successful. Communication with a central database may be necessary to validate any data input, but there is little other content in the application.

Challenge 2.7

Write down a quick PACT analysis for the introduction of a 'point of sale' system (i.e. where goods are priced and paid for) for a café at a motorway service station. Discuss your ideas with a colleague.

Summary and key points

The design of interactive systems is concerned with people, the activities they are undertaking, the contexts of those activities and the technologies that are used: the PACT elements. There is considerable variety in each of these and it is this variety – and all the different combinations that can occur – that makes the design of interactive systems so fascinating.

- The design of interactive systems requires the analyst/designer to consider the range of PACT elements and how they fit together in a domain.

- People vary in terms of physical characteristics and psychological differences and in their usage of systems.
- Activities vary in terms of temporal aspects, whether they involve cooperation, complexity, whether they are safety-critical and the nature of the content they require.
- Contexts vary in terms of physical, social, organizational aspects.
- Technologies vary in terms of the input, output, communication and content that they support.
- Undertaking a PACT analysis of a situation is a useful way of scoping a design problem.

Exercises

1 You have been asked to design the information system for a new cycle path network that is to run through part of your town. The aim of the system is to provide information on directions and distances for leisure cyclists to the main points of interest in the town. It also needs to provide information on other things, such as bus and train times for those cyclists who are commuting to and from work. Undertake a PACT analysis for this application.

2 For the same application produce a project development plan. You should detail what sort of requirements work will be needed to understand the domain, the people or skills that will be needed in the project team, and the approach that will be taken. Identify any milestones that you would have in the project.

Further reading

Norman, D. (1998) *The Design of Everyday Things.* Doubleday, New York. *Donald Norman discusses the ideas of mental models in several of his publications. This is probably the best.*

Getting ahead

Murrell, K.F.H. (1965) *Ergonomics – Man in his Working Environment*. Chapman & Hall, London.

Payne, S. (2012) Mental models. In J.A. Jacko (ed) *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 3rd edn. CRC Press, Taylor and Francis, Boca Ratun, FL.

Wilson, A. (2012) Sensor and recognition-based input for interaction. In J.A. Jacko (ed) *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 3rd edn. CRC Press, Taylor and Francis, Boca Ratun, FL.

Web links

The accompanying website has links to relevant websites. Go to
www.pearsoned.co.uk/benyon

Comments on challenges

Challenge 2.1

With VCRs came the video hire shop and so the activity of watching a film moved from the cinema into the home. VCRs also allowed films to be recorded from the television so people could watch them whenever they wanted. With DVDs people are given more options than just watching the film, so now the activity includes watching pieces that were cut out of the original film, slightly different versions, interviews with the actors and director and so on. The activity of watching a film is now more interactive: the people watching have more control over what they see.

Challenge 2.2

How an e-mail actually gets from one place to another is surprisingly complicated! It is much more like sending a letter by post than like making a telephone call. The e-mail is sent as one or more 'packets' of data which may be routed across the world by any of a host of different routes. The e-mail travels from your computer through the computer providing the Internet connection, then to a major 'hub' where it joins a high-capacity 'backbone' cable. As it comes closer to its destination this process is reversed as it moves off the main cables into more remote areas. A sophisticated database of addresses and routeing information is used to find the best way.

Challenge 2.3

Physically the siting is important so that people in wheelchairs, children, etc. can reach the buttons. Buttons must be easy to press so that the elderly are not excluded from their use. Psychologically the machine should not make undue demands on people. It is difficult to say anything certain since we do not know the complexity of the machine. Some ticket machines are very simple – designed just to select the destination and deliver the ticket. Others try to offer the whole range of functions, different ticket types, groups, period return tickets and so on. These machines tend to become very complicated and hard to use. From the usage perspective the design needs to support both those people who are in a hurry and perhaps use the machine every day and those people who have never encountered such a machine before, perhaps speak a different language and are trying to do something quite complex. It is difficult to design optimally for both of these types of use.

Challenge 2.4

Sending e-mails is a fairly frequent activity that is often interrupted. It is a straightforward activity in itself but it can become very complex when it is interleaved with other things such as finding old e-mails, finding addresses, attaching documents and so on. It is not necessary to coordinate the activity with others. The tasks of finding and entering addresses are made much easier if the e-mail program has an embedded address book, as the person only has to remember and type small amounts of data. Otherwise long e-mail addresses have to be typed in.

Challenge 2.5

For reasons of durability, we would suggest a touchscreen or ordinary monitor with tracker ball and a robust keyboard (for entering data such as name of hotel guest) or an on-screen version (rather tiresome to use). Other options are possible.

Challenge 2.6

The touchscreen used as an output device, plus a small printer embedded in the casing for confirmation of bookings, etc. would probably be more reassuring than just a confirmation number. Sound output (and indeed input) would be possible but is likely to be impractical in the noisy environment of the airport.

Challenge 2.7

There are many complex issues involved, of course. Here are just a few to start with. People - the whole range! From a coachload of football supporters or elderly people on an outing to individuals wandering around late at night. The key thing to consider is how to deal with crowds at one time and just a few people at another. The activities are simple and well defined. The items have to be identified, priced and totalled. The money has to be taken and a receipt printed. Occasionally there will be a question to be answered that goes outside this simple task structure, such as 'how much would this cost if I . . . ?', or disputes over prices need to be settled. There are also other stakeholders involved: the serving staff, the managers and so on. They also need information from the system. As for technologies, items could have a barcode on them, but for meals this is difficult, so usually individual items need to have the price typed in. This takes time. The interface design will be quite critical - e.g. there could be specific keys for things like tea and coffee, but whether it is a good idea to have a specific key for everything is another matter. Now you have had a chance to think about this, spend some time looking at all the different solutions that different cafes and restaurants use.

Chapter 3

The process of human-centred interactive systems design

Contents

Aims

Design is a creative process concerned with bringing about something new. It is a social activity with social consequences. It is about conscious change and communication between designers and the people who will use the system. Different design disciplines have different methods and techniques for helping with this process. Approaches to and philosophies of design change over time. In mature disciplines, examples of good design are built up and people can study and reflect upon what makes a certain design great, good or awful. Different design disciplines have different constraints, such as whether the designed object is 'stand-alone' or whether it has to fit in and live with legacy systems or conform to standards.

In this chapter we look at what is involved in interactive systems design and how to go about designing interactive systems. After studying this chapter you should be able to:

- Understand the nature of interactive systems design
- Understand the four processes involved in design: understanding, design, envisionment, evaluation
- Understand the centrality of evaluation in human-centred design
- Understand the scenario-based design approach
- Develop scenarios and personas
- Understand the scenario-based design method.

3.1 Introduction

There are many different ways of characterizing the activities involved in the design process. For David Kelley, founder of the product design company IDEO, 'Design has three activities: understand, observe and visualize'. He says:

> Remember, design is messy; designers try to understand this mess. They observe how their products will be used; design is about users and use. They visualize which is the act of deciding what it is.
> Kelley and Hartfield (1996), p. 156

In this chapter we provide methods and processes to help designers deal with the 'messy' problems of designing interactive systems. We characterize the overall design process in terms of the four activities illustrated in Figure 3.1. The key features of this representation are as follows:

- Evaluation is central to designing interactive systems. Everything gets evaluated at every step of the process.
- The process can start at any point – sometimes there is a conceptual design in place, sometimes we start with a prototype, sometimes we start with understanding.
- The activities can happen in any order, for example understanding might be evaluated and a prototype built and evaluated and some aspect of a physical design might then be identified.

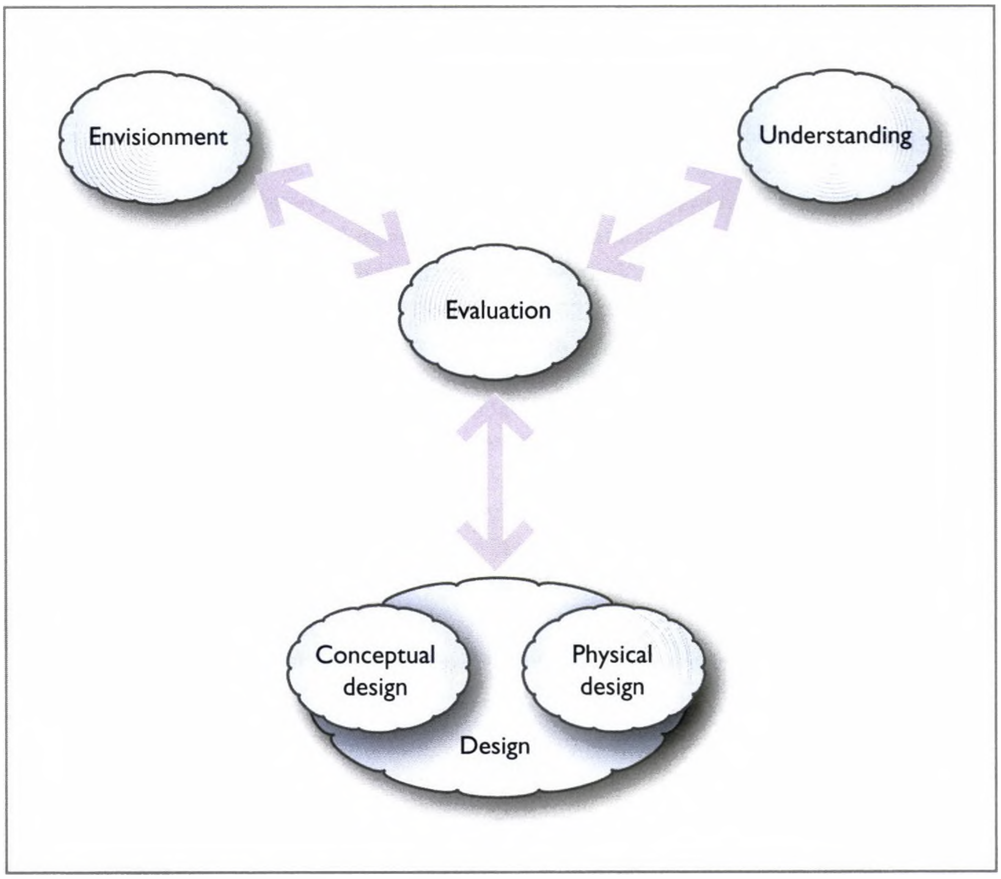

Figure 3.1 Understanding, design, evaluation, envisionment

Understanding

Understanding is concerned with what the system has to do, what it has to be like and how it has to fit in with other things: with the requirements of the product, system or service. Designers need to research the range of people, activities and contexts relevant to the domain they are investigating so that they can understand the requirements of the system they are developing. They need to understand the opportunities and constraints provided by technologies.

There are both functional and non-functional requirements to consider. Functional requirements are concerned with what the system should be able to do and with the functional constraints of a system. It is important for the designer to think about the whole interaction experience in an abstract way. Deciding who does what, when something should be displayed or the sequence in which actions are undertaken should come later in the design process. A good analysis of an activity will strive to be as independent of current practice as possible. Of course, there are always functional constraints – the realities of what is technically possible – which render certain ordering, sequencing and allocation of functions inevitable. There are also logical and organizational constraints that may make particular designs infeasible.

→ Chapter 7 gives a detailed treatment of methods for understanding

Requirements are generated through discussions and interactions with people who will use or be affected by the proposed system – the stakeholders (see Box 3.1). Requirements are also generated through observations of existing systems, research into similar systems, what people do now and what they would like to do. Requirements can be generated through working with people in focus groups, design workshops and so on, where different scenarios can be considered (see Section 3.4). The aim is to collect and analyse the stories people have to tell. Requirements are essentially about understanding.

BOX 3.1

Stakeholders

'Stakeholders' is a term that refers to all the people who will be affected by any system that results from the process of interactive systems design. This includes the people who will finish up using the new system (sometimes called the 'users'), but it also includes many other people. For example, the organization that the system is being designed for will probably have many people in it that will not be using the system, but will be affected by it as it might change their job. For example, introducing a new website into an organization often changes working practices as well as simply providing information. There may be stakeholders outside the organization, such as government authorities, that need to verify some procedures. The number and type of people affected by a new interactive system will vary greatly according to what sort of system it is. An important part of the understanding process is to consider all the different stakeholders and how they might be affected, to decide who should be involved in discussions about the design.

Design

Design activities concern both conceptual design and physical design. Conceptual design is about designing a system in the abstract, physical design is concerned with making things concrete.

Conceptual design

Conceptual design is about considering what information and functions are needed for the system to achieve its purpose. It is about deciding what someone will have to know to use the system. It is about finding a clear conceptualization of a design solution and how that conceptualization will be communicated to people (so that people will quickly develop a clear mental model).

← Chapter 2 discusses mental models

There are a number of techniques to help with conceptual design. Software engineers prefer modelling possible solutions with objects, relationships and 'use cases' (a semi-formal scenario representation). Entity–relationship models are another popular conceptual modelling tool. Flow can be represented using dataflow diagrams and structure can be shown with structure charts. The conceptual design of a website, for example, will include a site map and a navigation structure.

→ Methods for modelling are discussed in Chapter 9

One way to conceptualize the main features of a system is to use a 'rich picture'. Two examples are shown in Figure 3.2. A rich picture captures the main conceptual relationships between the main conceptual entities in a system – a model of the structure of a situation. Peter Checkland (Checkland, 1981; Checkland and Scholes, 1999), who originated the soft systems approach, also emphasizes focusing on the key transformation of a system. This is the conceptual model of processing. The principal stakeholders – customers, actors, system owners – should be identified. The designer should also consider the perspective from which an activity is being viewed as a system (the *Weltanschauung*) and the environment in which the activities take place. (Checkland proposes the acronym CATWOE – customers, actors, transformation, *Weltanschauung*, owners, environment – for these elements of a rich picture.) Most importantly, the rich picture identifies the issues or concerns of the stakeholders, thus helping to focus attention on problems or potential design solutions.

The key feature of conceptual design is to keep things abstract – focus on the 'what' rather than the 'how' – and to avoid making assumptions about how functions and information will be distributed. There is no clear-cut distinction between conceptual and physical design, but rather there are degrees of conceptuality.

Physical design

Physical design is concerned with how things are going to work and with detailing the look and feel of the product. Physical design is about structuring interactions into logical sequences and about clarifying and presenting the allocation of functions and knowledge between people and devices. The distinction between conceptual and physical design is very important. The conceptual design relates to the overall purpose of the whole interactive system. Between the people and the technologies there has to be enough knowledge and ability to achieve the purpose. Physical design is concerned with taking this abstract representation and translating it into concrete designs. On one side this means requirements for hardware and software and on the other it defines the knowledge required by people and the tasks and activities that people will have to do. There are three components to physical design: operational design, representational design and design of interactions.

Operational design is concerned with specifying how everything works and how content is structured and stored. Taking a functional view of an activity means focusing on processes and on the movement, or flow, of things through a system. *Events* are occurrences that cause, or trigger, some other functions to be undertaken. Sometimes these arise from outside the system under consideration and sometimes they arise as a result of doing something else. For example, some activity might be triggered on a particular day or at a particular time; another might be triggered by the arrival of a person or document.

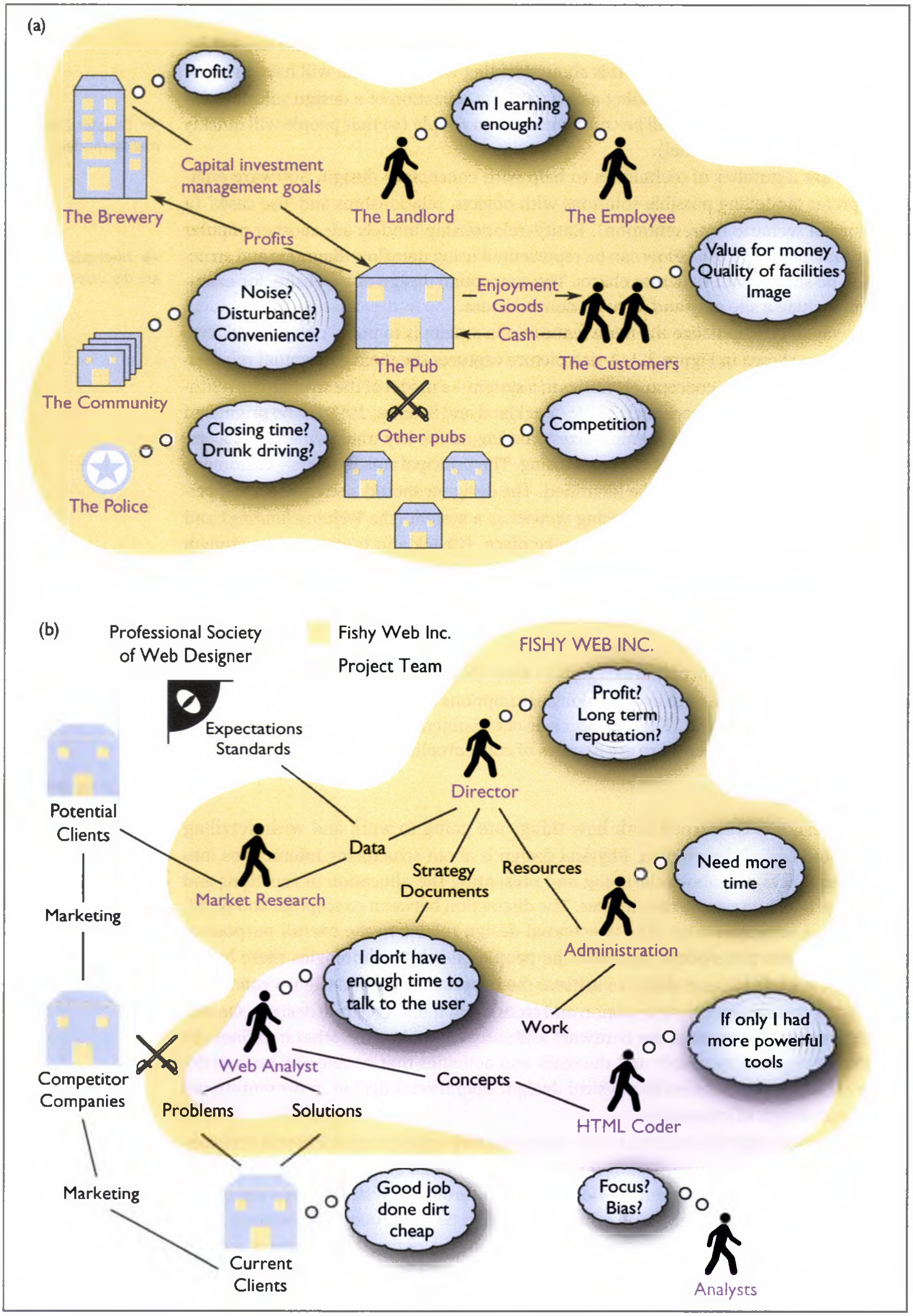

Figure 3.2 Rich pictures of (a) a pub and (b) a Web design company

(Source: After Monk, A. and Howard, S. (1998) Methods & Tools: the rich picture: a tool for reasoning about work context, *Interactions*, 5(2), pp.21–30. © 1998 ACM, Inc. Reprinted by permission)

Representational design is concerned with fixing on colours, shapes, sizes and information layout. It is concerned with style and aesthetics and is particularly important for issues such as the attitudes and feelings of people, but also for the efficient retrieval of information.

→ See Section 12.5 on information design

Style concerns the overall 'look and feel' of the system. Does it appear old and 'clunky' or is it slick, smooth and modern? What mood and feelings does the design engender? For example, most Microsoft products engender an 'office' and 'work' mood, serious rather than playful. Many other systems aim to make the interaction engaging, some aim to make it challenging and others entertaining. In multimedia and games applications this is particularly important.

Interaction design, in this context, is concerned with the allocation of functions to human agency or to technology and with the structuring and sequencing of the interactions. Allocation of functions has a significant impact on how easy and enjoyable a system is to use. Designers create tasks for people by the way they allocate functions.

For example, consider the activity of making a phone call. Conceptually speaking, certain functions are necessary: indicate a desire to make a phone call, connect to the network, enter the phone number, make connection. Years ago a telephone exchange was staffed by people and it was these people who made connections by physically putting wires into connectors. In the days of wired phones, picking up the receiver indicated the desire to make a call, the full number had to be dialled in and then the telephone exchange would automatically make the connections. Nowadays a person just has to press the connect button on a cellular phone, choose someone's name from the phone's address book and the technology does the rest.

Recall the activity–technology cycle (see Chapter 2). The allocation of knowledge and activities between people and technologies is a significant part of how experiences change over time.

Challenge 3.1

Find a colleague and discuss the activity of watching pre-recorded films on TV. Focus on the way the allocation of function changes with the technology such as VCR, DVD and PVR (personal video recorder). How has it changed now that on-line films are easily available on your TV or PC?

Envisionment

Designs need to be visualized both to help designers clarify their own ideas and to enable people to evaluate them. Envisionment is concerned with finding appropriate media in which to render design ideas. The medium needs to be appropriate for the stage of the process, the audience, the resources available and the questions that the designer is trying to answer.

→ Chapter 8 presents techniques for envisionment

There are many techniques for envisionment, but they include any way in which abstract ideas can be brought to life. Sketches 'on the back of an envelope', fully functioning prototypes and cardboard mock-ups are just some of the methods used. Scenarios, sometimes represented in pictorial form as storyboards, are an essential part of prototyping and envisionment. They provide a way of working through a design idea so that the key issues stand out. Scenarios are discussed below.

Evaluation

Evaluation is tightly coupled with envisionment because the nature of the representation used will affect what can be evaluated. The evaluation criteria will also depend on who is able to use the representation. Any of the other design activities will be followed by an evaluation. Sometimes this is simply the designer checking through to make sure something is complete and correct. It could be a list of requirements or a high-level design brief that is sent to a client, an abstract conceptual model that is discussed with a colleague, or a formal evaluation of a functional prototype by the future system users.

→ Chapter 10 provides detail on evaluation

Techniques for evaluation are many and various, depending once again on the circumstances. Expressing the design ideas in terms of a concrete scenario that people have to work their way through can be very effective. The important thing to keep in mind is that the technique used must be appropriate for the nature of the representation, the questions being asked and the people involved in the evaluation.

Challenge 3.2

If you were to have a new room built onto your house – or have a room converted from one use to another – consider the processes that you would have to go through, starting with:

- *A conceptual design*
- *A physical design*
- *Some requirements*
- *A prototype or other envisioned solution.*

Implementation

Figure 3.1 does not include the implementation or production of the design (nor all the planning and management stages of a project). But, of course, ultimately things have to be engineered and software has to be written and tested. Databases have to be designed and populated and programs have to be validated. The whole system needs to be checked to ensure that it meets the requirements until finally the system can be formally 'launched' and signed off as finished. Since this book is primarily about design, we do not spend a lot of time on issues of implementation, but they can account for a significant portion of total development costs. Clients will often want extra features when they see a system nearing completion, but these will have to be costed and paid for. On the other hand, the developers need to ensure that their system really does meet the specification and does not contain any 'bugs'.

→ Chapter 9 provides a number of semi-formal models

If interactive systems designers were architects they would have well-understood methods and conventions for specifying the results of the design process. They would produce various blueprints from different elevations and engineering specifications for particular aspects of the design. In interactive systems design there are a variety of formal, semi-formal and informal methods of specification. The best known of the formal methods is the Unified Modeling Language (UML) (Pender, 2003).

BOX 3.2

Agile development

Over the past few years there has been a move away from large software engineering approaches to the development of interactive systems towards 'agile' development methods. These are designed to produce effective systems of high quality that are fit for purpose, but without the huge overhead associated with the planning and documentation of a large IT (information technology) project.

There are a number of competing methods, but probably the best known comes from DSDM, a not-for-profit consortium of software development companies. Their system, called Atern, is fully documented, showing how software can be developed in small teams. There is still plenty of debate about how well these methods, such as extreme programming (Beck and Andres, 2004), fit in with human-centred approaches, but many of the methods do promote participation between developers and stakeholders. In particular, Obendorf and Finck (2008) describe a method bringing together agile methods and scenario-based design.

3.2 Developing personas and scenarios

In order to guide the design process, designers need to think about the PACT elements (introduced in Chapter 2). The people who will use the system are represented by personas: profiles of the different types, or archetypes, of people the designer is designing for. Activities and the contexts in which they will occur are envisioned through scenarios of use. Different concrete scenarios can be used to envision how different technologies could function to achieve the overall purpose of the system. Personas and scenarios are developed through the understanding process, using any of a wide range of methods (discussed in Chapter 7), and through undertaking a PACT analysis. Almost inevitably, personas and scenarios evolve together as thinking about people involves thinking about what they want to do, and thinking about activities involves thinking about who will be undertaking them!

Personas

Personas are concrete representations of the different types of people that the system or service is being designed for. Personas should have a name, some background and, importantly, some goals and aspirations. Alan Cooper introduced the idea of personas in the late 1990s (Cooper, 1999) and they have gained rapid acceptance as a way of capturing knowledge about the people the system or service is targeted at. In the latest edition of his book (Cooper *et al.*, 2007), he links personas very closely with his ideas of goal-directed design. Personas want to be able to do things using your system. They want to achieve their aims, they want to undertake meaningful activities using the system that the designer will produce. Designers need to recognize that they are not designing for themselves. Designers create personas so that they can envisage whom they are designing for. They create personas so that they can put themselves in other people's shoes.

As any new system is likely to be used by different types of people, it is important to develop several different personas. For example, in designing a website for people interested in the author Robert Louis Stevenson (described in more detail in Chapter 14), we developed personas for a school teacher in Germany, a university lecturer from the UK, a child in Africa and a Stevenson enthusiast from the USA. Such a diverse group of people have very different goals and aspirations, and differ in all the ways – physically, psychologically and in terms of the usage they would make of the site (see Chapter 2).

Scenarios

Scenarios are stories about people undertaking activities in contexts using technologies. They appear in a variety of forms throughout interactive systems design and are a key component of many approaches to design.

Scenarios have been used in software engineering, interactive systems design and human–computer interaction work for many years. More recently, scenario-based design has emerged as an important approach to the design of interactive systems in the twenty-first century (Alexander and Maiden, 2004).

One of the main proponents of scenario-based design is John Carroll, and his book *Making Use* (2000) remains an excellent introduction to the philosophy underlying the approach. In it he illustrates how scenarios are used to deal with the inherent difficulty of doing design. Drawing on the activity–technology cycle (Figure 2.1) to show the position in product development, he argues that scenarios are effective at dealing with five key problems of design (Figure 3.3):

- The external factors that constrain design such as time constraints, lack of resources, having to fit in with existing designs and so on.
- Design moves have many effects and create many possibilities, i.e. a single design decision can have an impact in many areas and these need to be explored and evaluated.

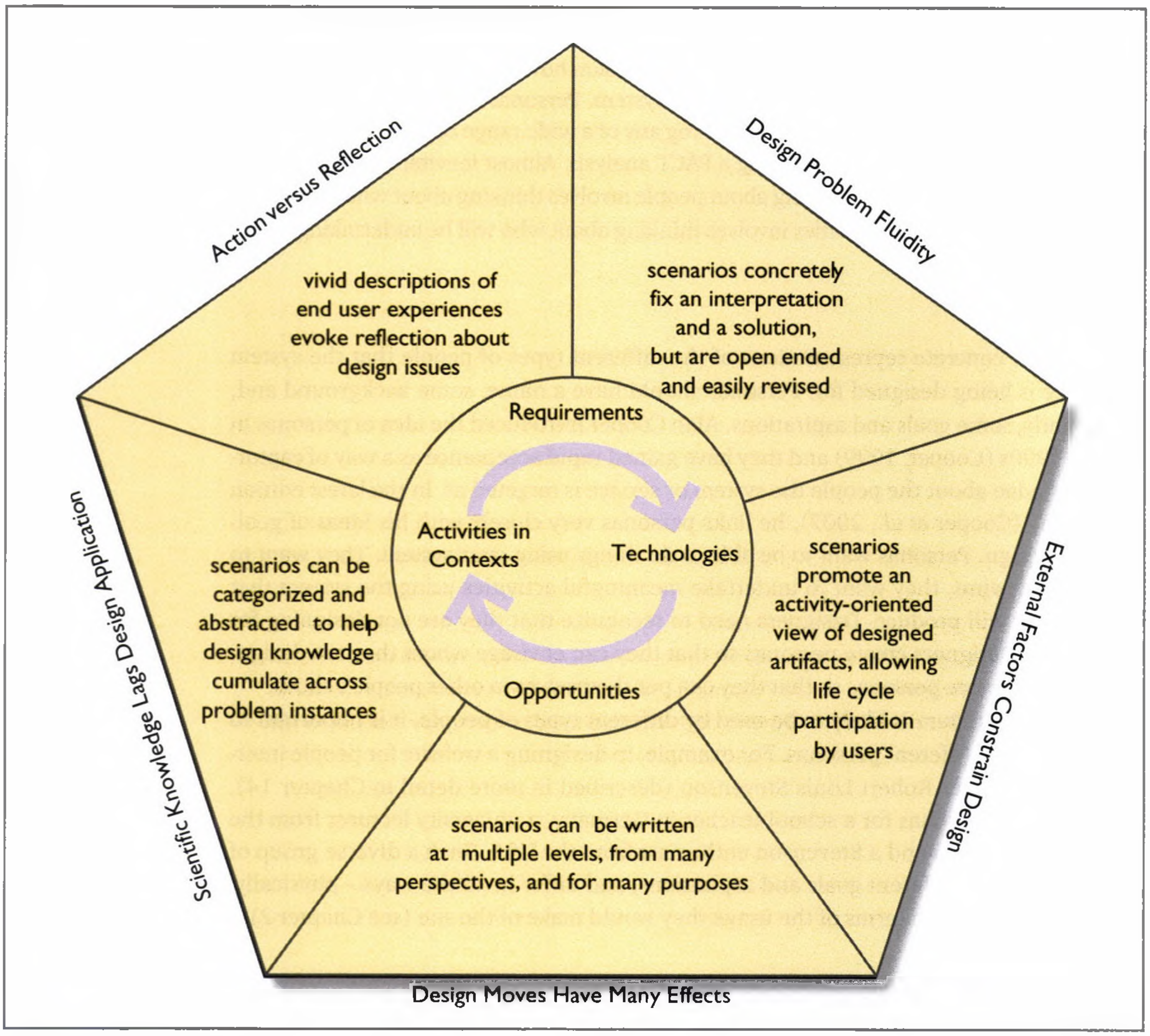

Figure 3.3 Challenges and approaches in scenario-based design

(Source: After John M. Carroll. *Making Use: Scenario-based Design of Human-Computer Interactions.* Fig. 3.2, p. 69 © 2000 Massachusetts Institute of Technology, by permission of the MIT Press)

- How scientific knowledge and generic solutions lag behind specific situations. This point concerns generalities. In other design disciplines, general design solutions to general design problems have evolved over the years. In interactive systems design this does not happen because the technology changes as soon as, or even before, general solutions have been discovered.
- The importance of reflection and action in design.
- The slippery nature of design problems.

Our scenario-based design method is presented in the next section. Below are a few examples of how we have used personas and scenarios in a recent project. Some are quite detailed, others are single snapshots of interactions used to explore design options.

Example: Companions

We have recently been looking at a novel form of interaction that goes under the title of 'companions'. Companions are seen as an intelligent, personalized, multimodal interface to the Internet. Companions know their 'owners' and adapt the interaction to personalized interests, preferences and emotional state. In investigating the companions concept we have developed a number of persons and scenarios.

A health and fitness companion (HFC), for example, would help provide advice and companionship for people in the domain of health and fitness. We explored the idea in a two-day workshop attended by a number of the project partners. During and subsequent to this workshop, three personas were developed to explore the various needs of people with differing lifestyles, levels of fitness and exercise regimes. These are shown in Figures 3.4, 3.5 and 3.6.

One central theme of the explorations concerned the motivational approaches that would be suitable for different scenarios and personas. The Sandy persona (Figure 3.4), for example, would need more encouragement and persuasion to exercise than the Mari persona (Figure 3.5), perhaps by preventing a recorded television programme from being shown until training is completed. Another aspect, concerning social networking, was explored through the Bjorn persona (Figure 3.6). Thus the personas were developed to reflect particular design issues and values. The whole issue of persuasion technologies is a difficult one for interaction design.

BOX 3.3

Captology

B. Fogg introduced the idea of persuasion technologies, or 'captology' as he terms it, in the late 1990s. It is a controversial idea. The basic aim of captology is to persuade people to do things they otherwise would not do. At first sight this looks somewhat immoral. Who are we, as designers, to persuade people to do something they don't want to do? However, we can see examples, such as the Sandy persona, where persuading him to exercise is for his own good. We also need to persuade people to take precautions if things are dangerous. I am quite happy that a software system persuaded me to save my work before the system crashed (on the other hand, why did the system not just save it for me?). Persuasion is 'a non-coercive attempt to change attitudes or behaviors of people' (Fogg *et al.* 2007). However, if this is persuading me to buy something that I cannot afford, then it is not good at all, whether it is non-coercive or not. This is an area of HCI where ethics and values must be taken seriously.

In another exploration we looked at the concept of a companion to deal with digital photos. Such a companion would be functional in helping organize, edit and share photos, but would also be a conversational partner. We envisaged a companion that could discuss photos with its owner and perhaps reminisce about events and people.

Sandy

- age 46
- drives a lot
- drinks and eats too much
- recently divorced
- children in early 20s
- had recent health scare (suspected heart attack which was actually angina)
- kids have bought him a HFC

1. We meet Sandy in a hospital room, he's being visited by his kids.
2. They are worried about his health, he does little exercise and since his wife left him his diet has become appalling.
3. They give him a HFC (what is this?!) which will combine with his current home system. They explain that it's intended to help raise his general level of fitness, monitor his health and set and maintain a healthy balanced diet.
4. They all leave the hospital and Sandy starts the configuration.
5. Being ex-army Sandy decides that a tough-love drill instructor personality would suit him best (he's on board with the fact that he needs to get healthy), so he selects Alf, a no-nonsense archetype companion character.
6. He opens his exercise regime to be accessible by his children, on their request, as he feels this will be an added incentive for him to exercise.
7. Configuration involved biometrics such as weight, height, etc., allowing Alf to suggest appropriate training and diet.
8. It's aim is to understand whether the owner is in bad condition needing to get better, wanting to maintain current health or aim for high performance.
9. Alf reprimands bad behaviour (such as buying unhealthy food), nags when he doesn't exercise, but offers positive motivation when he does.

Figure 3.4 The Sandy persona for HFC scenario

Mari

- age 23
- aerobics instructor
- training seriously for first marathon
- her usual training partner has moved away
- she leads a wild social life and tends to burn the candle at both ends
- she's got a targeted schedule
- companion is very proactive in pace making and motivation

1. She's set up a long-term schedule with her HFC to enable her to run her first marathon in under 4 hours.
2. This includes target goals such as what times she should be running long distances by which stage of the regime.
3. The HFC adapts to maintain the regime when Mari's social circumstance impacts her ability to train.
4. If she runs too far or too fast the companion will advise that this may have a negative impact on her training and may result in potential injury.
5. Explicit instructions in real time run ('ok, now we're gonna push hard for 2 minutes....ok, well done, let's take it easy for the next 5....etc.').
6. The HFC has access to her social schedule (through social companion?) and suggests going to a party the night before a long run may not be a great idea.
7. At the actual marathon her HFC becomes a motivating force and gives her real-time advice (eg, 'there's a hill coming up, pace yourself', it knows this from a run plug-in she bought for the HFC).

Figure 3.5 The Mari persona for HFC scenario

Bjorn

- Age 32
- Office worker (ad account manager)
- No children, lives alone
- Dog died (used to walk it for exercise)
- Starting to put on weight
- Used to play football at university, much less active now
- Active social life
- 'I want to stay fit, but on my own time and fitting in to my own schedule'

1. Home from work, he was meant to go out the previous evening but got invited out to a dinner party instead. This evening is now free, so he decides to go for a run.
2. He's in his living room and sets up for his run. This involves:
   - route choice
   - exercise level, eg easy jog or hard run (specific pacing feedback choice, eg within PB)
   - music choice
   - disturbability status (eg, open to contact/running partner)
   - weather
   - (warm up/stretching?)
3. He gets changed and leaves the house, the handover is transparent from living room companion to mobile device-based companion and is aware of all Bjorn's choices regarding run setup.
4. Just as he's about to begin, the sun breaks through the clouds and Bjorn decides he'd rather go for a longer run than initially selected in his living room; this change must be facilitated through his mobile companion device. Selective rather than creative process (eg, chose run three on route 2).
5. He starts running hard.
6. Asked whether he's warmed up as he's running above a warm-up rate.
7. He slows down to a more gentle jog and reaches his start point.
8. A touch of the device indicates he's starting his run.
9. Music begins.
10. Pace-setting tactile feedback begins.
11. Midway through run he's informed that Julie is also running in the woods and has set her HFC at open to running partners (this is a closed list of the pre-set social network that Bjorn belongs to).
12. He slows down and runs on the spot and sends her a greeting, asking if she'd like to join him; she says yes.
13. She catches up and the companion automatically reconfigures his pacing settings to match hers.
14. After a circuit they part ways and Bjorn heads home.
15. On entering the house Bjorn warms down and stretches which induces a brief summary on his mobile device whilst the detailed data from his run is transparently transferred to his home network.
16. He walks into the kitchen to grab a glass of water and plan what to make for dinner. His home companion notes that he went for a long run today so he must be hungry, and suggests some recipes based on what he has in his fridge: 'how about the steak, it goes out of date tomorrow'. Nothing takes his fancy so he asks the companion to search online whilst he has a shower. Takes shower, comes down and is presented with some new recipes and the fact that Julie called and asked him for a drink that night.
17. At a later time he asks for an overview of his past three months' exercise. His companion notes that his heart rate is recovering quicker which suggests he's getting fitter, but for the past two weeks he's not been running for as long.

Figure 3.6 The Bjorn persona for HFC scenario

Figure 3.7 illustrates a scenario in which a person has a large collection of photographs and wishes to search for a specific image from a recent trip. One feature of this scenario was to explore different modalities for the companion. The interaction employs both speech and touch depending on the activity being undertaken. For example, it is much quicker to specify specific search parameters through speech than by typing or clicking a series of check boxes (part 2 in the scenario). However, when it comes to flicking through the search-generated group or applying certain other editorial functional tasks such as scaling and cropping, touch

1. The user is moving from a standard view of their photos to a search mode. This is a voice driven function.

2. Here the user narrows down the field by establishing a search parameter again by voice. Note that the user could search for any metadata parameter or combination of parameters that the system has established. Indeed the system could proactively suggest additional ones.

3. Having used voice to establish the smaller field, the user now applies touch to quickly flick through the pictures. Additional touch functionality could include scaling, cropping or editing.

4. Having found the photo they want to send, the user now combines speech with touch to indicate that the gesture of flicking to the left means email that specific image to the user's uncle.

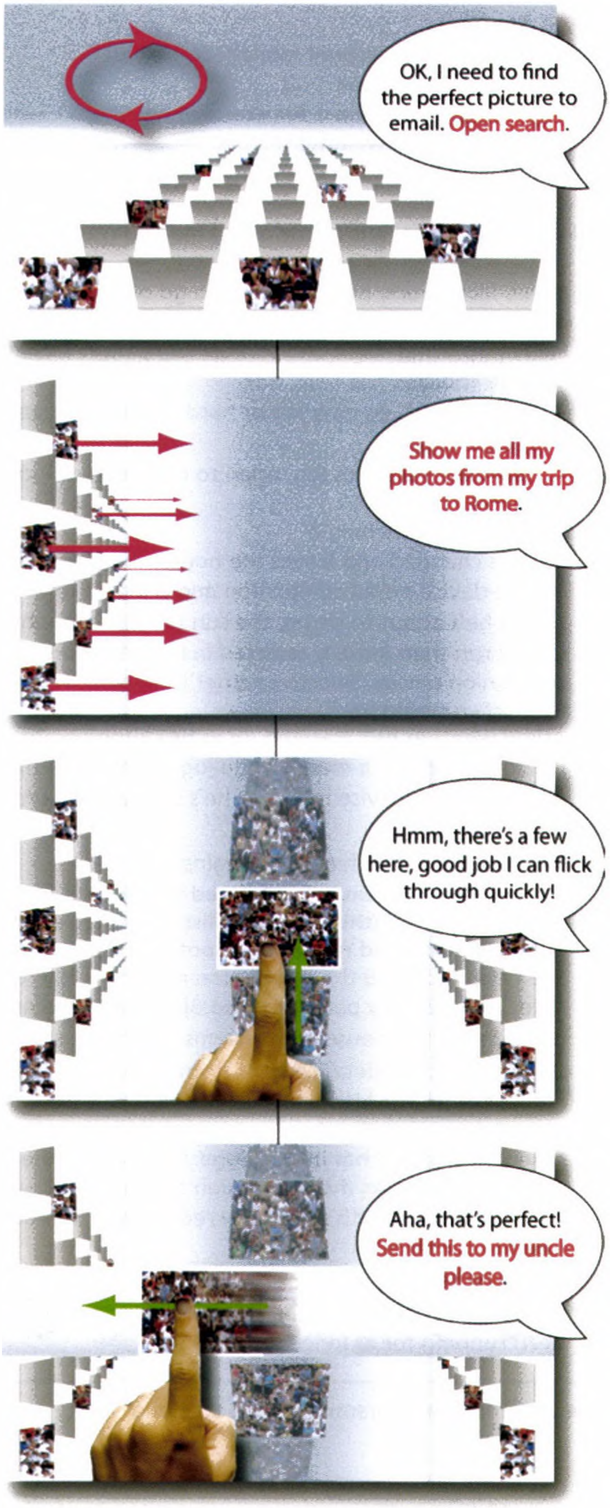

Figure 3.7 An scenario of multimodal interaction with a photo companion

becomes the more natural interaction. For example, it's quicker to drag a finger back and forth to resize an image in a serendipitous or haphazard fashion than it is to say, 'Make that image a little bigger . . . bit bigger . . . bit bigger . . . no, that's too big . . . bit smaller . . . too small' and so on. However, for specific categorial edits speech may be best, for example 'Make this image 4 by 6 inches and print'. The true power of the interaction experience comes from the considered use of both in conjunction.

In another scenario we were looking at environmental influence on the interaction. For example, Figure 3.8 shows the potential for moving between displays. Small displays (e.g. digital photoframes) have a more limited touch capability than a larger display (in the case of Figure 3.8 an interactive coffee table). Figure 3.9 illustrates a further option, namely that of using a display that is simply too far from the person to be touched. This in many ways most fairly reflects the current living-room environment. In such a situation physical gesture becomes an appropriate option, either by using one's hands or by wielding an object, such as is used in the Nintendo Wii games console. This allows for parameters such as speed, direction and shape of movement.

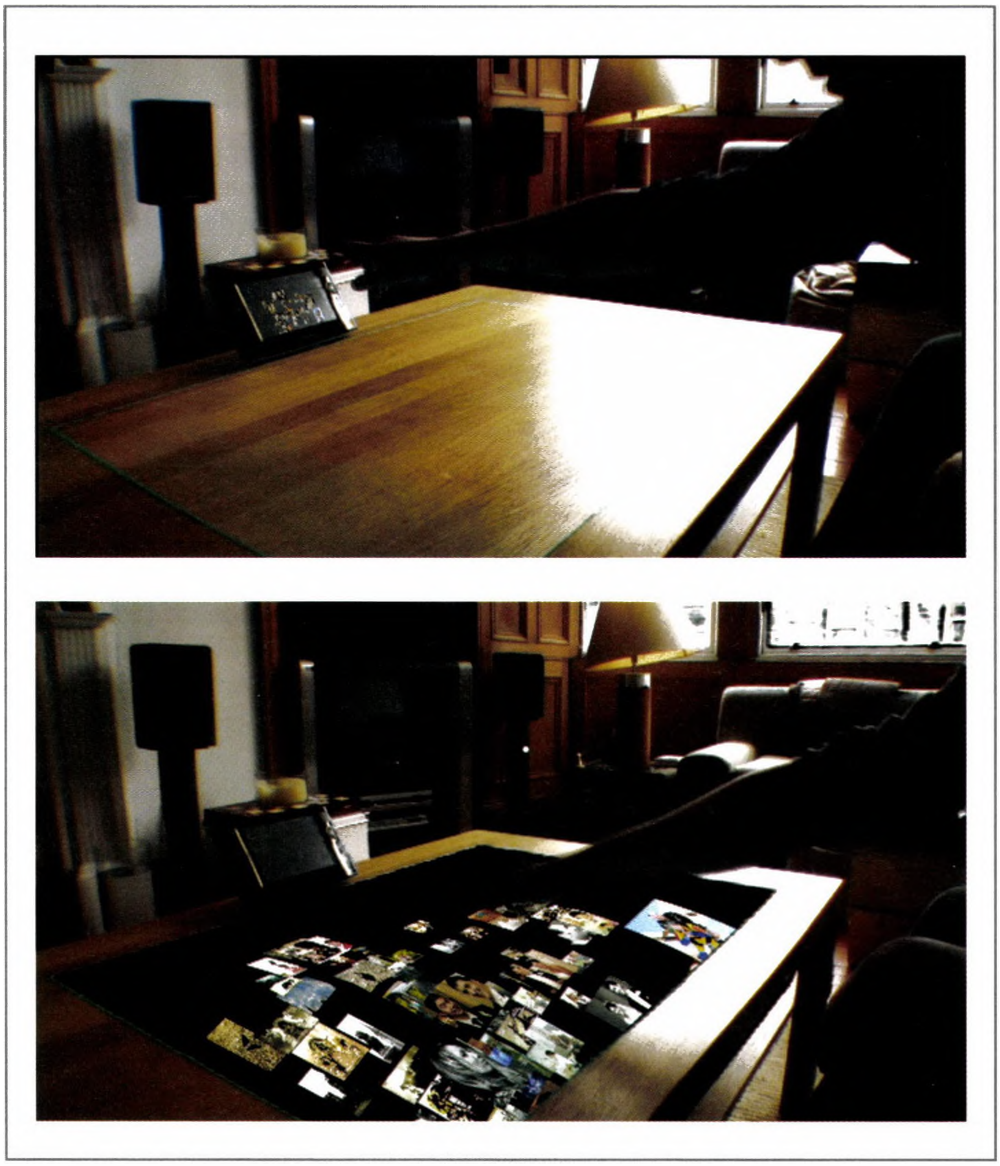

Figure 3.8 An example of a multimodal interaction moving between displays from a digital photoframe to a smart coffee table

Figure 3.9 An example of a gesture-based multimodal interaction with a remote screen

3.3 Using scenarios throughout design

Scenarios (and their associated personas) are a core technique for interactive systems design. They are useful in understanding, envisioning, evaluation, and both conceptual and physical design: the four key stages of interactive system design (Figure 3.1). We distinguish four different types of scenario: stories, conceptual scenarios, concrete scenarios and use cases. Stories are the real-world experiences of people. Conceptual scenarios are more abstract descriptions in which some details have been stripped away. Concrete scenarios are generated from abstract scenarios by adding specific design decisions and technologies and, once completed, these can be represented as use cases. Use cases are formal descriptions that can be given to programmers. At different stages of the design process, scenarios are helpful in understanding current practice and any problems or difficulties that people may be having, in generating and testing ideas, in documenting and communicating ideas to others and in evaluating designs.

The place of the different types of scenario and the processes and products of the design process are illustrated in Figure 3.10. The lines joining the types of scenario indicate the relationships between them. Many stories will be represented by a few conceptual scenarios. However, each conceptual scenario may generate many concrete scenarios. Several concrete scenarios will be represented by a single use case. The difference between these types is elaborated below.

Figure 3.10 also illustrates three critical processes involved in design and how they interact with the different scenario types. Designers abstract from the details of stories to arrive at conceptual scenarios. They specify design constraints on conceptual scenarios to arrive at concrete scenarios. Finally they formalize the design ideas as use cases.

Stories

➜ Chapter 7 discusses techniques for getting stories

Stories are the real-world experiences, ideas, anecdotes and knowledge of people. These may be captured in any form and comprise small snippets of activities and the contexts in which they occur. This could include videos of people engaged in an activity, diary entries, photographs, documents, the results of observations and interviews and so on. People's stories are rich in context. Stories also capture many seemingly trivial details that are usually left out if people are asked to provide more formal representations of what they do.

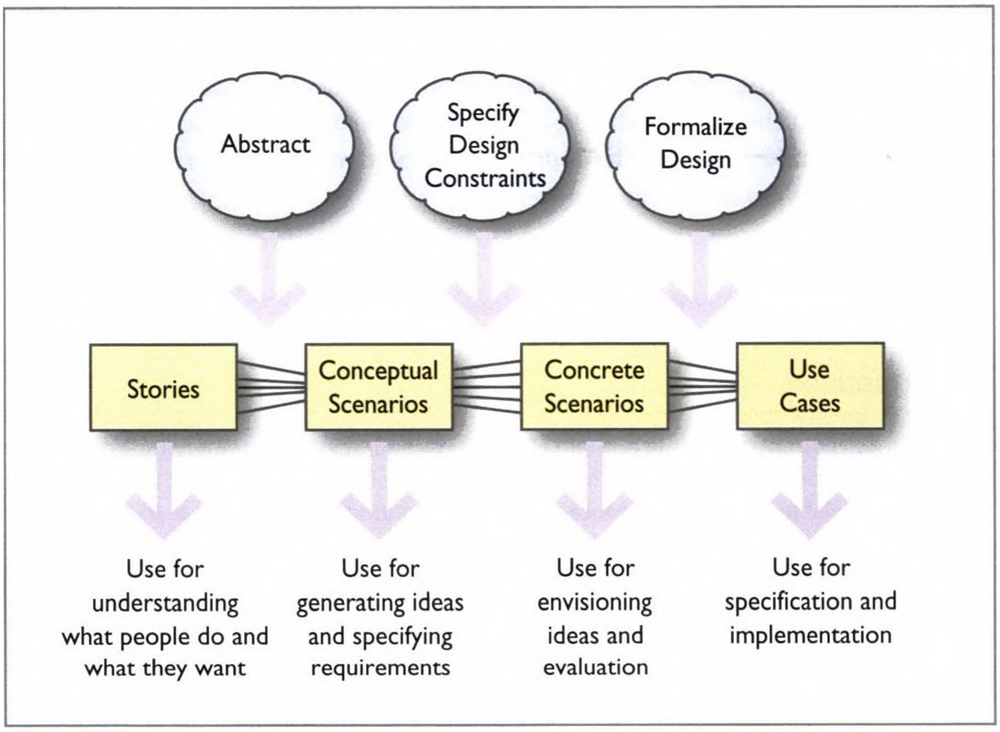

Figure 3.10 Scenarios throughout design

Example

Here is a story from someone describing what happened last time he made an appointment to see his local doctor.

> *'I needed to make an appointment for Kirsty, my little one. It wasn't urgent – she had been having a lot of bad earache every time she had a cold – but I did want to see Dr Fox since she's so good with the children. And of course ideally it had to be when Kirsty was out of school and I could take time off work. I rang the surgery and the receptionist told me that the next appointment for Dr Fox was the next Tuesday afternoon. That was no good since Tuesday is one of my really busy days, so I asked when the next one was. The receptionist said Thursday morning. That meant making Kirsty late for school but I agreed because they sounded very busy – the other phone kept ringing in the background – and I was in a hurry myself. It was difficult to suggest a better time without knowing which appointments were still free.'*

Conceptual scenarios

Conceptual scenarios are more abstract than stories. Much of the context is stripped away during the process of **abstraction** (see Box 3.4) and similar stories are combined. Conceptual scenarios are particularly useful for generating design ideas and for *understanding* the requirements of the system.

Example

Once the designer has accumulated a collection of stories, common elements will start to emerge. In this case a number of stories such as the one above result in the conceptual scenario below, describing some requirements for a computerized appointments system.

> **Booking an appointment**
>
> People with any degree of basic computer skills will be able to contact the doctors' surgery at any time via the Internet and see the times which are free for each doctor. They can book a time and receive confirmation of the appointment.

As you can see, at this stage, there is little or no specification of precise technologies or how the functions will be provided. The scenario could be made more abstract by not specifying that the Internet should be used or more concrete (that is, less abstract) by specifying that the booking should be made from a computer rather than a mobile phone. Finding an appropriate level of abstraction at which to describe things for a given purpose is a key skill of the designer.

BOX 3.4

Abstraction

The process of abstraction is one of classification and aggregation: moving from the details of specific people undertaking specific activities in a specific context using a particular piece of technology to a more general description that still manages to catch the essence of the activity.

Aggregation is the process of treating a whole thing as a single entity rather than looking at the components of something. In most domains, for example, one would aggregate a screen, processor, disk drive, keyboard and mouse and treat this as a single thing - a computer - rather than focusing on the components. However, in another situation one of the components - processor speed, or disk size, say - may prove to be critical and so it would be better to have two aggregations: fast computers and slow computers, say.

Classification is the process of recognizing that things can be collected together, so that dealing with the class of things is simpler (more abstract) than dealing with the individual things. There are no set ways to classify things, so the analyst has to work with the stories that have been gathered and with the people themselves to decide which things belong together and why.

Between them, aggregation and classification produce abstractions. Of course, there are different degrees of abstraction and it is one of the skills of a designer to settle upon an appropriate level. The most abstract level is to treat everything simply as a 'thing' and every activity as 'doing something', but such an abstract representation is not usually very useful.

Concrete scenarios

Each conceptual scenario may generate lots of concrete scenarios. When designers are working on a particular problem or issue they will often identify some feature that applies only under certain circumstances. At this point they may develop a more specific elaboration of the scenario and link it to the original. Thus one reasonably abstract scenario may spawn several more concrete elaborations which are useful for exploring particular issues. Notes that draw attention to possible design features and problems can be added to scenarios.

Concrete scenarios also begin to dictate a particular interface design and a particular allocation of functions between people and devices. Concrete scenarios are particularly useful for prototyping and envisioning design ideas and for evaluation because they are more prescriptive about some aspects of the technology. However, there is not a clean break between conceptual and concrete scenarios. The more specific the scenario is about some aspects, the more concrete it is.

Example

In this example, decisions have now been taken concerning drop-down menus, the fact that the next two weeks' details are to be shown, and so on. However, the notes following the scenario show that there are many design decisions still to be taken.

Booking an appointment/01

Andy Dalreach needs a doctor's appointment for his young daughter Kirsty in the next week or so. The appointment needs to be outside school-time and Andy's core working hours, and ideally with Dr Fox, who is the children's specialist. Andy uses a PC and the Internet at work, so has no difficulty in running up the appointments booking system. He logs in [1] and from a series of drop-down boxes, chooses to have free times for Dr Fox [2] displayed for the next two weeks [the scenario would continue to describe how Andy books the appointment and receives confirmation].

Notes to booking an appointment/01

1 Is logging in necessary? Probably, to discourage bogus access to the system, but check with the surgery.
2 Free times can be organized by doctor, by time of day, or by next available time. Drop-down boxes will save screen space but may present problems for less experienced users or those with poor eyesight.

Use cases

A use case describes the interaction between people (or other 'actors') and devices. It is a case of how the system is used and hence needs to describe what people do and what the system does. Each use case covers many slight variations in circumstances – many concrete scenarios. The lines in Figure 3.10 indicate how many concrete scenarios result, after the process of specification and coding, in a few use cases.

Before use cases can be specified, tasks and functions have to be allocated to humans or to the device. The specification of use cases both informs and is informed by the task/function allocation process. This is the interaction design part of physical design.

Finally, all the design issues will be resolved and the set of concrete scenarios is then used as the basis of the design. A set of use cases can be produced which specifies the complete functionality of the system and the interactions that will occur. There are a number of different ways of representing use cases – from very abstract diagrams to detailed 'pseudo code'. Figure 3.11 shows the 'booking an appointment' use case in a typical format.

→ See also Chapter 11 on task analysis

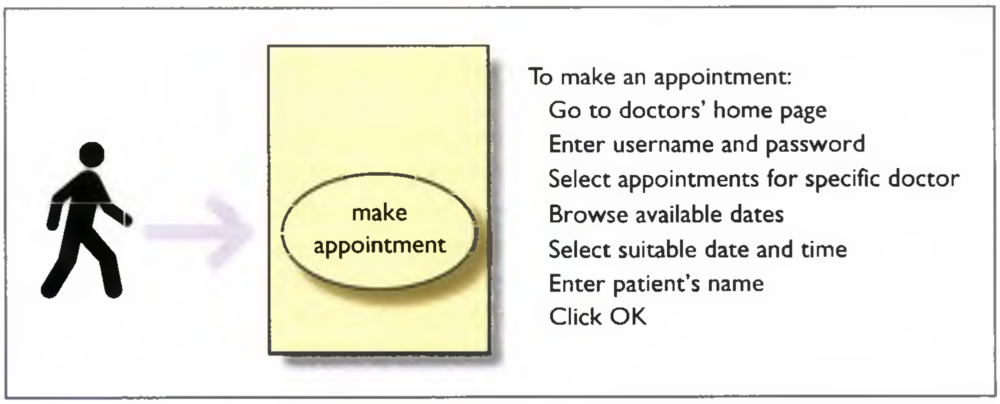

Figure 3.11 Use case for booking an appointment

BOX 3.5

Use cases

Despite the fact that use cases have been a core element of software engineering methods since the late 1980s, the concept remains elusive and different authors define a use case in different ways. In a section called 'use cases undefined', Constantine and Lockwood (2001) rage against the lack of clear definition for such a critical term. The definition used in the Unified Modeling Language (UML) – an attempt to provide commonly agreed specification concepts and notation for software engineering – is too lengthy and obscure to repeat here. Constantine and Lockwood also point out that how the use case is specified – in a sort of pseudo programming code as we have done, or simply using the diagrammatic ellipse and named role as some do, or otherwise – also varies considerably between authors and methods.

It is also the case that use cases are used at different levels of abstraction. Constantine and Lockwood's 'essential use cases' are similar to the conceptual scenarios described here and there are others who base a whole design method on use case modelling. We reserve the term 'use case' for describing an implementable system, i.e. enough interface features have been specified, and the allocation of functions between people and the system has been completed, so that the use case describes a coherent sequence of actions between an actor and a system. The term 'actor' is used here because sometimes we need to specify use cases between one part of the system (a 'system actor') and another, but usually the actor is a person.

Challenge 3.3

Find a vending machine or other relatively simple device and observe people using it. Write down their stories. Produce one or more conceptual scenarios from the stories.

3.4 A scenario-based design method

The use of the different types of scenario throughout design can be formalized into a scenario-based design method. This is illustrated in Figure 3.12 with, once again, products of the design process shown as boxes and processes shown as clouds. Besides the four different types of scenario, four other artefacts are produced during the design process: requirements/problems, scenario corpus, object model and design language. The specification of a system is the combination of all the different products produced during the development process.

Each of the main processes – understanding, envisionment, evaluation and design – is the subject of a chapter in the next part of the book. An important thing to notice is the relationship between specifying design constraints and the use of scenarios. For envisionment and most evaluation, the scenarios have to be made more concrete. This means imposing design constraints. However, this does not mean that the designer needs to design a new physical, concrete scenario each time he or she wants to envision a possible design. It may be that designers imagine a scenario with particular design constraints imposed and this helps them evaluate the design. This sort of 'what if?' generation and evaluation of concrete scenarios is a common and key aspect of design.

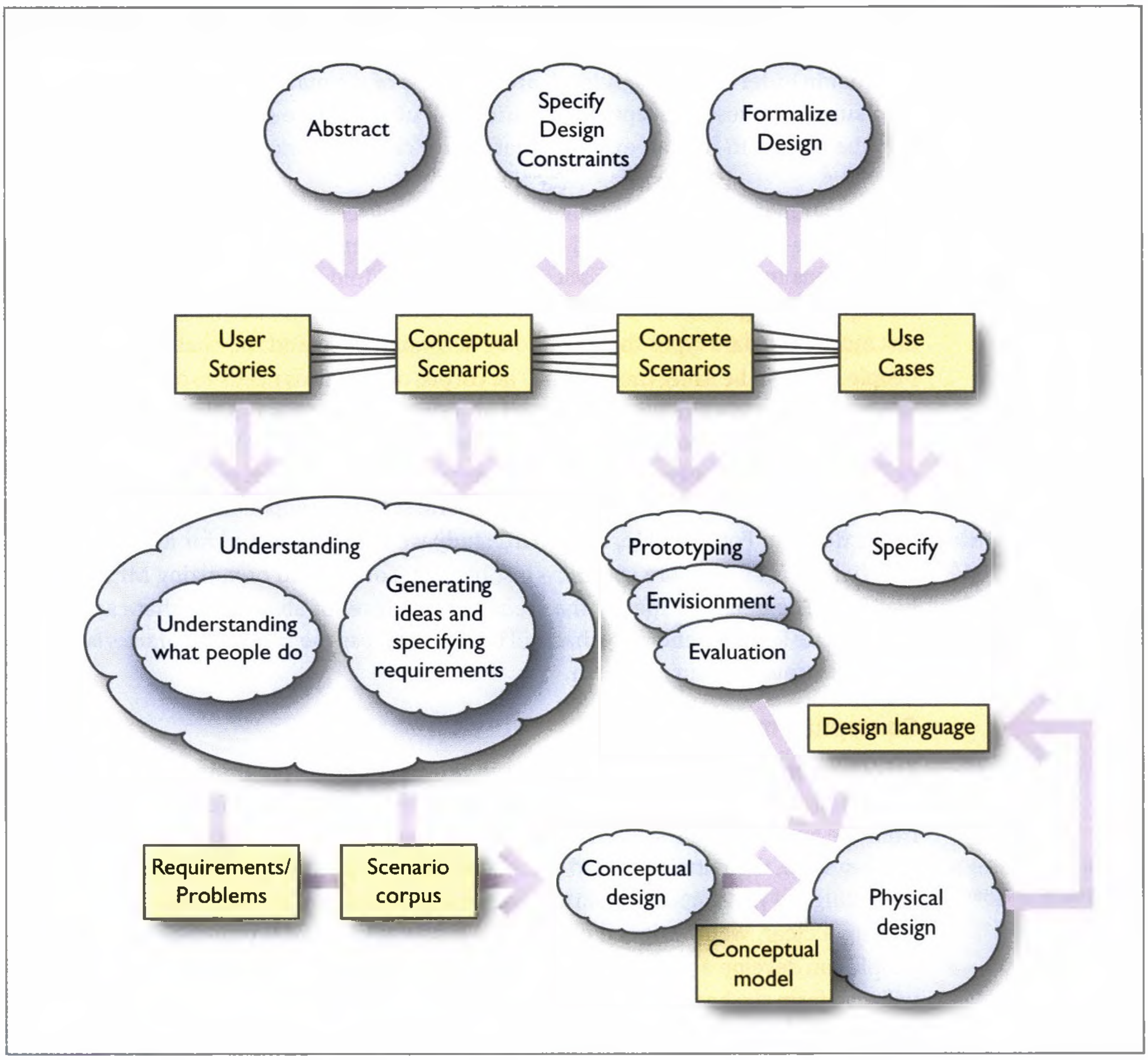

Figure 3.12 Overall scenario-based design method

The key products that have not been discussed so far are: requirements and problems; scenario corpus; conceptual model; and design language. These are briefly introduced below for completeness, but a full understanding will require more in depth study.

→ Further details are given in Chapters 7–10.

Requirements and problems

In the gathering of people's stories and during the analysis and abstraction process, various issues and difficulties will come to light. These help the analyst/designer to establish a list of requirements – qualities or functions that any new product or system should have. For example, in the HFC example, the companion had to be available both at home and when exercising. It needed information about routes and personal preferences. The requirements and problems product is a prioritized list of issues that the system to be designed needs to accommodate.

→ See Chapter 7 on understanding for requirements

Scenario corpus

In our approach we seek to develop a representative and carefully thought-through set, or corpus, of scenarios. Having undertaken some analysis activities designers will have gathered a wide range of user stories. Some of these will be very general and some will be quite specific. Some will be fairly simple, straightforward tasks; others will be more vague. It is important at some point for the designer to pull these disparate experiences together in order to get a high-level, abstract view of the main activities that the product is to support. These conceptual scenarios will often still be grounded in a real example; the trick is to find an example that shares characteristics with a number of other activities.

The rationale for the development of a corpus of scenarios is to uncover the 'dimensions' of the design situation and to demonstrate different aspects of those dimensions. Dimensions include characteristics of the various domains within which the product will operate (e.g. large and small domains, volatile or static domains, etc.), the various media and data types that need to be accommodated and the characteristics of the people who will be using the system. The corpus of scenarios needs to cover all the main functions of the system and the events that trigger the functions. Different types of interaction need to be present along with any key usability issues. The dimensions include different types of content and how that can be structured, issues of style and aesthetics.

→ The HIC case study is in Chapter 6

A corpus of scenarios might consist of several scenarios depending on the complexity of the domain. For example, in the HIC study we had eleven, and for an MP3 application (which is much more specific – just playing, sorting and organizing MP3 files) we had five. The aim is to specify the scenarios at a level of abstraction that captures an appropriate level of generality that will be useful across the range of characteristics that is demonstrated within a domain.

Conceptual model

→ Conceptual modelling is covered in Chapter 9

An object or data model results from the process of conceptual modelling, including developing the scenarios and understanding the objects and actions that are evident from the analysis of the scenario corpus. The conceptual model shows the main objects in the system, their attributes and the relationships that exist between them. Conceptual modelling is a very important part of interactive systems design that is often overlooked. Having a clear, well-designed conceptual model will make it easier to design so that people can develop a good, accurate mental model of the system. The conceptual model will also form the basis of the information architecture of a system and of any metaphor that is used in the design.

Design language

→ We return to this in Chapter 9

The design language produced consists of a set of standard patterns of interaction and all the physical attributes of a design – the colours, shapes, icons and so on. These are brought together with the conceptual actions and objects, and the 'look and feel' of the design is completed. A 'design language' defines the key elements of the design (such as the use of colour, style and types of buttons, sliders and other widgets, etc.) and some principles and rules for putting them together. A consistent design language means that people need learn only a limited number of design elements and then they can cope with a large variety of different situations.

Challenge 3.4

Take a look at the operating system that you use on your computer and identify some key elements of the design language that is used.

Documenting scenarios

Scenarios can become messy, so in order to control the scenarios a structure is needed. We use the PACT framework (people, activities, contexts, technologies) to critique scenarios and to encourage designers to get a good description of the scenario. For each scenario the designer lists the different people who are involved, the activities they are undertaking, the contexts of those activities and the technologies that are being used. We also structure scenario descriptions. Each scenario should be given an introduction. The history and authorship can be recorded, along with a description of how the scenario generalizes (across which domains) and the rationale for the scenario. Each paragraph of each scenario should be numbered for ease of reference and endnotes included where particular design issues are raised. Endnotes are particularly useful in documenting issues raised during the development of the scenario. They are a way of capturing the claims being made about the scenarios (Rosson and Carroll, 2002). Examples of relevant data and media should be collected.

← Chapter 2 describes PACT

BOX 3.6

Trade-offs and claims analysis

Rosson and Carroll (2002) describe an approach to scenario-based design in which scenarios are used throughout the design process and how they help designers to justify the claims that they make about design issues. Design is characterized by trade-offs. There is rarely a simple solution to a problem that solves all the issues. Usually the adoption of one design will mean that something else cannot be achieved. Designers need to document their design decisions so that the trade-offs can be evaluated. Scenarios help by making the rationale for the design explicit. Designers can record the claims that they make about their designs. Claims analysis is an important part of scenario-based design and is used in identifying problems or in thinking through possible future designs (Rosson and Carroll, 2002). The process is simply to identify key features of a scenario and to list good and bad aspects of the design. Rosson and Carroll use a technique of putting a '+' beside good features and a '–' beside bad features. Claims analysis makes the rationale behind a design explicit.

A similar method is to list the design questions, design options and criteria used to make choices, the QOC method (MacLean *et al.*, 1991).

Challenge 3.5

Take a device or system that you have to hand – a mobile phone, a website, a vending machine – and critique the design, focusing on the aspects that are central to its use. Make a list of claims about the design.

When working in a large design team, it is useful to accompany scenarios with real data. This means that different team members can share concrete examples and use these as a focus of discussion. Another key feature of writing scenarios is to think hard about the assumptions that are being made: to make assumptions explicit or deliberately avoid making things explicit in order to provoke debate. In particular, the use of personas can help to focus on specific issues. For example, an elderly woman with arthritis might be one of the personas, thus foregrounding issues of access and the physically impaired interacting with technology.

Finally, with scenarios it is important to provide a very rich context. The guiding principles for scenario writing are people, activities, contexts and technologies.

Example: Scenario MP3/01 – 'How does that song go again?'

This example illustrates how scenarios can be structured and used to think about designs and become part of a corpus. The context for this scenario was the development of a Home Information Centre (HIC). The HIC was envisaged as a new type of information, communication and entertainment device that would look good in the home and, whilst providing similar functions to a computer, would have a novel interface making it far more enjoyable and natural to use.

→ The HIC case study is in Chapter 6

In developing the MP3 player function for the HIC, we explored a number of different scenarios, finally finishing with five that defined the MP3 function corpus. The example here shows the scenario being used to explore requirements and concepts for the HIC. Notice that whilst being quite abstract, it is concrete enough to bring design issues to the fore. Figure 3.13 shows a QOC claims analysis for this scenario.

SCENARIO MP3/01

Title
'How does that song go again?'

Scenario type
Activity scenario

Overview
People = Anne, a single female, computer-literate. Works at home.
Activities = Searching for MP3 tracks.

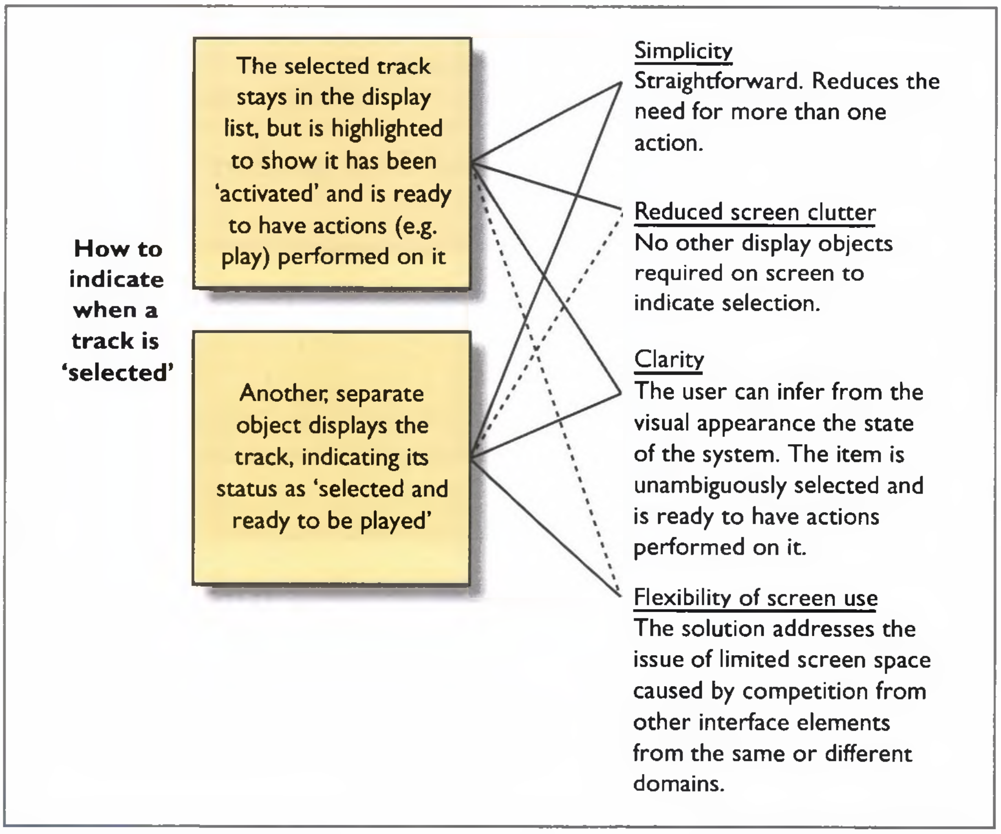

Figure 3.13 An example of the QOC (questions, options, criteria) claims method

Context = Apartment with office/study space where user works from home. HIC is in the kitchen, which is adjacent to the study.

Technology = The HIC and a PC.

Rationale

The substantive activity here is the use of the search function to find a specific MP3 track. The use of different search parameters is described. The user interrogates the HIC using keyboard input; future elaborations might deal with other modalities such as voice input.

P1. Anne is a freelance arts journalist who works mainly from home. She's writing a piece for a national newspaper about singer-songwriters, and is irritated to find she can't remember the lyrics of a particular well-known song she wants to quote in her article. She knows the name of the singer and the song title, but beyond that, her memory is failing her.

P2. She leaves her desk, hoping a break and a cup of coffee will dispel the block. While in the kitchen, she decides she'll check the HIC for new messages [1]. While she is doing this, she realizes the HIC's MP3 player [2] can help her out.

She remembers she has downloaded the song [3] she needs at some time in the past two months or so, and knows it's still in the HIC's memory [4].

P3. She selects the 'play' function (Level 1 of the HIC's actions bar) [5], which takes her down one level in the HIC interface, to where she can see 'MP3 search' [6]. She selects this and the familiar Rolodex interface comes up, asking her to enter some search details. She can search by entering an artist name, track title or music genre – these are all elements of an MP3 track's identity which the HIC can recognize. She is about to enter the artist's name, but realizes she has stored several of this singer's tracks in the HIC; to get a unique search result first time, she enters the track name instead, using the keyboard [7].

P4. The HIC quickly finds the track and asks her [8] if she wants to play it now. She does, and selects this option by touching the screen [9]. The MP3 controller appears on the screen, with the selected track already loaded and ready to play.

P5. She touches the 'play' button and listens. She increases the volume [10]. The lyrics she wants come back to her straight away – she can now go back to her desk. She leaves the HIC on [11] (without thinking).

P6. Later on, she has finished her piece and e-mailed it to her editor. But she wants to hear the song again, as it has sentimental memories for her. Fortunately, she has left the HIC on in MP3 mode. All she needs to do is select 'replay' and the song plays again.

P7. She decides she has put in enough time at the computer for the day, and feels like watching some TV. She chooses the TV device on the HIC and settles down to watch the early evening news [12].

Notes to scenario MP3/01

1. Checking messages is peripheral to the MP3 domain, but it is interesting to consider how MP3-related activities fit in with other domains of the HIC. Multiple screen objects will soon start to compete with each other for screen space.
2. 'MP3 player' is meant here in the general sense of the MP3 domain – that is, all the functions relating to MP3.
3. How she has done this is not described here – but see scenario MP3/02 for a more detailed account of this activity.
4. The question of how the HIC stores MP3 and other files is a significant one. One of the popular features of the MP3 format is the ease with which files can be shuffled from one platform to another; this will involve frequent use of saving, copying, deleting and other functions. This may imply the need for some sort of 'File Manager' function in the HIC (cf. scenarios MP3/02, /03, /04).
5. The Actions Bar is now a well-established part of the HIC prototype. Here, Anne goes one level down the HIC's navigation hierarchy to get to the MP3 domain, and her point of entry is the 'play' icon, found on the Actions Bar. But there may be other points of entry too – say, from a 'Favourites' menu or similar.

6 The MP3 domain may be made up of different modules – a 'player' with functional controls, a search function, track lists, and so on. Some or all of these may be present on screen at one time; this raises the question of what the 'default' configuration will be: only the basic functional controls? All the different modules? And how will the user call these up or dismiss them as required?
7 Consider other modalities too: handwritten using a stylus and pressure pad? Voice input?
8 How is the search result presented to Anne? It may be in the form of a list, with the results containing the parameters she gave the HIC. The search may return several results, and there should be a way for her to unambiguously select a track from a list. This could be tricky unless the text is at a good size and spacing for touching on a screen – unless some other selection method is used.
9 She is close to the screen – but could she select the option remotely too?
10 Perhaps the HIC could sample the level of background noise in the area, and adjust the playback volume automatically.
11 Is there a screen saver?
12 What happens to the MP3 interface when the TV comes on? Presumably the whole of the HIC's information space will be filled by the TV image. Other functions and controls will need to be cleared from the screen (perhaps returning automatically when the TV is turned off). Or they could be reduced greatly in size, and put onto a display bar on the periphery of the screen. Perhaps there could be a 'bring to front' command (operated remotely, or by voice?) to make other controls available while the TV was still active.

Cross-referencing scenario types

Another aspect of documentation that is useful is to cross-reference the stories to the conceptual scenarios, through the concrete examples and finally to the use cases. A simple Web-based system can be developed, as illustrated in Figure 3.14.

Other researchers have suggested similar ideas that capture the multiple views necessary to see how scenarios and claims work together to provide a rich understanding of how a design finished as it did.

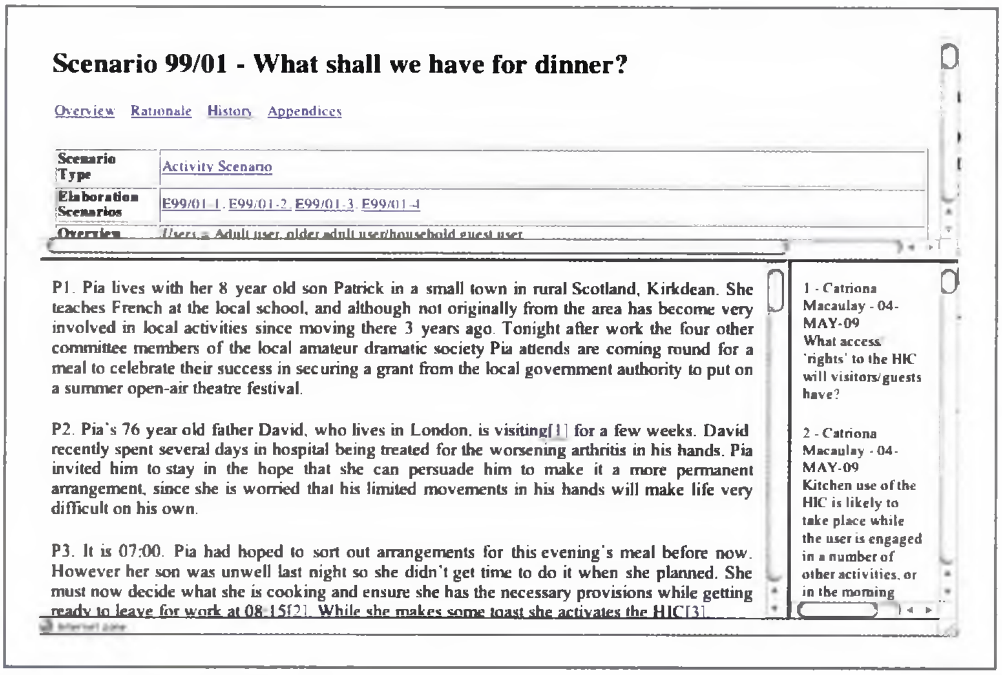

Figure 3.14 The scenario web

Summary and key points

The design of interactive systems is concerned with people, the activities they are undertaking, the contexts of those activities and the technologies that are used. This chapter has introduced the main elements of design - understanding, envisionment, design and evaluation - and how scenario-based design and the development of personas can be used to guide the designer. Scenarios and their different uses in this process have been explored.

- Scenarios are stories about the interactions between people, activities, contexts and technologies.
- Scenarios offer an effective way of exploring and representing activities, enabling the designer to generate ideas, consider solutions and communicate with others.
- Scenarios are used throughout the design process and, along with the requirements and problems, conceptual design and design language can form part of the specification of the system.

Exercises

1. Find someone using some technology to do something and observe what they do. Now write down the story associated with that activity. Abstract a conceptual scenario from this one experience, by removing the contextual detail and other details about the specific interface to the technology. Now think of an alternative design for a device that would allow someone to undertake a similar activity and generate a concrete scenario based on these design constraints. Finally, specify this as a use case.
2. Develop a scenario corpus for people using a vending machine. Consider the dimensions of the usage, the contexts for the interaction and the range of people that you would want to consider.

Further reading

Cooper, A., Reiman, R. and Cronin, D. (2007) ***About Face 3: The Essentials of Interactive Design***. **Wiley, Hoboken, NJ.** *Cooper et al. give an insightful and enjoyable tour through some of the worst aspects of interactive systems design and introduce their approach, which focuses on developing personas and taking a goal-oriented approach to design.*

Rosson, M.-B. and Carroll, J. (2012) Scenario-based design. In J.A. Jacko (ed.) ***The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications***, **3rd edn.** CRC Press, Taylor and Francis, Boca Raton, FL.

Winograd, T. (ed.) (1996) ***Bringing Design to Software***. **ACM Press, New York.** *This book contains a number of interesting articles from interactive systems designers and is essential reading for all would-be interactive systems designers.*

Getting ahead

Interactions is an excellent journal focusing on interactive systems design.

Carroll, J.M. (ed.) (1995) ***Scenario-based Design.*** Wiley, New York.

Carroll, J.M. (2000) ***Making Use: Scenario-based Design of Human–Computer Interactions.*** MIT Press, Cambridge, MA.

Rosson, M.-B. and Carroll, J. (2002) ***Usability Engineering.*** Morgan Kaufmann, San Francisco, CA.

John (Jack) Carroll has been hugely influential in the area of human–computer interaction over many years and with his wife, Mary-Beth Rosson, has written extensively on scenario-based design. The first of his books is a collection of papers showing how the scenario concept appears in a variety of guises throughout human–computer interaction and software engineering. The second is compiled from many of his writings and presents a thoughtful and coherent approach to developing systems using scenarios. It illustrates how scenarios are appropriate in all the stages of systems development. The third is a practical design book.

Web links

There are some good white papers on Jared Spools' website, some of which cover personas. See **www.uie.com**

The accompanying website has links to relevant websites. Go to **www.pearsoned.co.uk/benyon**

Comments on challenges

Challenge 3.1

When watching a movie on a VCR, the TV would often automatically select the appropriate TV channel and the VCR would start playing automatically. Thus the function of 'start film' was allocated to the device, and often the function of 'select VCR channel' was allocated to the device. With DVDs the human often has to select the appropriate channel and has to select 'start film', or 'play movie', from a menu. So people now have extra tasks to perform. Moreover, the default option on DVDs is often not 'play movie', so the person has to navigate to the appropriate option, giving them even more tasks. PVRs are different again, requiring people to undertake several tasks to watch a film. With huge numbers of films, clips and longer parts of films, YouTube requires people to undertake much more searching and selecting. When allocating functions to people or to devices, think hard about what tasks you are forcing people to undertake.

Challenge 3.2

A conceptual design would focus on the idea for the room. You may think it would be nice to have a conservatory or a downstairs toilet and proceed from there. You would evaluate the idea, perhaps by looking at some physical prototypes at a large store or at a friend's house. This might help to define the requirements, such as the size of the conservatory, where you would locate it and so on. Starting with a physical design, you might see something at a friend's or on television and this might trigger the idea that it would be a nice thing to have. Once you have the concept, proceed as above. Seeing a picture in a book is another example of an envisioned solution starting the process off. On other occasions the process might be started by some requirements. You may feel that you need a study, a new room for a baby, or somewhere to sit in the sun in the winter and it might be these requirements that begin the process. Notice how, wherever the process starts from, the next step will be some evaluation.

Challenge 3.3

A man wearing an overcoat and carrying a backpack came up to the machine and stared at it for two or three minutes. Whilst he was doing this two younger men came up behind him and were trying to look over his shoulder. Finally, he put his hand in his pocket and inserted some money. He pressed two buttons, B and 7, and watched as a packet of crisps was deposited in the tray.

You can imagine a few more stories such as this, resulting in a conceptual scenario along the lines of 'A person comes up to the machine, studies the instructions and what is available, inserts money, presses two buttons and retrieves the goods'.

Challenge 3.4

Key aspects of the design language are standard features of things such as windows and the different types of windows (some that are resizable, some that are not, etc.). Other features include the design of menus, dialogue boxes, alert boxes and so on. The colours are also consistent and chosen to evoke different feelings in people.

Challenge 3.5

Of course, this will depend on the device you have chosen and on how you approach the critique. The design principles (Chapter 4) are a good way to think about designs. A critique of a vending machine, for example, might include the claims:

✓ Useful for out-of-hours sales
X Limited selection of goods
✓ Quick interaction
X Does not always give change
X Mis-operation results in lengthy and time-consuming complaints
✓ High service costs.