Only study guide for

INF1520

Human-Computer Interaction 1

edited by

HELÉNE GELDERBLOM

Contributors:

Heléne Gelderblom

Paula Kotzé

Chris Johnson

Ruth de Villiers

School of Computing

University of South Africa, Pretoria

Printed and published by the
University of South Africa
Muckleneuk, Pretoria

Page make-up by
The School of Computing

INF1520/1/2012-2018

Table of Contents

Preface

The purpose of the study of Human-Computer Interaction is:

1. To enhance the quality of the interaction between human and machine and to systematically apply knowledge about human purposes, capabilities, and limitations, as well as knowledge about machine capabilities and limitations.
2. To develop or improve productivity and the functionality, safety, utility, effectiveness, efficiency, and usability of systems that include computers (Preece et al., 2007)

We therefore make a study of HCI to determine how we can make computer technology more usable for people. This requires the understanding of the following:

- The computer technology involved.
- The people who interact with the computer technology.
- The design of interactive systems and interfaces that are usable.
- The broader impact of computer technology on society and on our social, personal and working environment.

These four strands form the focus of this module. We look at the history of interactive systems. Human cognitive and physical capabilities are discussed and how knowledge of these should feed into the design of computer technology. Design problems, guidelines and tools are discussed, as well as methods for evaluating the designs. Lastly we discuss the broader social issues and consequences of advanced computer technology for all humans.

UNIT 1
Introduction to Human-Computer Interaction

CONTENTS

UNIT 1 - OUTCOMES

After studying this unit you should:

- Understand the historical context of HCI:
  - Where did HCI innovations and philosophy come from?
  - Who were the major people involved?
  - What were the important systems?
  - How did the ideas move from the laboratory to the market?
- Be able to define what HCI is.
- Be able to define important concepts related to HCI.
- Understand the current context of HCI.
- Be able to describe future directions of the field.
- Understand why HCI is considered to be a multidisciplinary subject and list the related disciplines.

1.1 Introduction

Computers and computer software are created for people to use. They should therefore be designed in a way that allows the intended user to use them successfully for the intended purpose and with the least amount of effort. To design a successful system the designers must know how to support the tasks that the user will perform with it. They must understand why the users need the system, what tasks they will want to perform with the system, what knowledge they might have (or lack) that may influence their interaction with the system, and how the system fits into the user's existing context.

The term *human-computer interaction* (HCI) was adopted in the mid-1980s to denote a new field of study concerned with studying and improving the effectiveness and efficiency of computer use. Today it is a multi-disciplinary subject with computer science, psychology and cognitive science at its core (Dix et al., 2004). When HCI became one of the domains of cognitive science research in the 1970s, the idea was to apply cognitive science methods to software development (Carroll, 2003). General principles of perception, motor activity, problem solving, language and communication were viewed as sources that could guide design. Although HCI has now expanded into a much broader field of study, it is still true that knowledge of cognitive psychology can help designers to understand the capabilities and limitations of the intended users. Human perception, information processing, memory and problem solving are some of the concepts from cognitive psychology that are related to people's use of computers (Dix et al., 2004).

We return to cognitive psychology and its role in HCI in Unit 2. In this unit we explain the historical context within which HCI developed, the current context within which it is practiced, and we provide some definitions of 'human-computer interaction' and related concepts.

1.2 The Historical Context

To put the development of interactive computing devices into context, we start this unit with an overview of the history of computing and HCI.

1.2.1 The Middle Ages

The early history of computing can be traced back to the narrow aims of mathematicians, logicians, and astronomers who had particular calculations that needed to be performed. The Persian astrologer Al-Kashi (1393-1449) built a device to calculate the conjunction of the planets. Records of this work survived and were transported to Europe, although the device itself was lost. The German mathematician, Wilhelm Schickard (1592-1635) developed a much less sophisticated tool to perform simple addition and subtraction. The Schickard machine was destroyed during the 30 Years War. The French mathematician Blaise Pascal (1612-1662) was forced to replicate much of Schickard's work but only succeeded in building an even more simplified version of that machine.

There was no gradual improvement in our knowledge over time. War, famine, and plague interrupted the development of mechanical computing devices. This, combined with the primitive nature of the hardware, meant that user interfaces were almost nonexistent. The systems were used by the people who built them.

There was little or no incentive to improve HCI.

1.2.2 The Eighteenth and Nineteenth Century

The agricultural and industrial revolutions in Western Europe created the need for external markets and external sources of raw materials. This greatly increased the level of trade that was already conducted in commodities such as spices, gold, and slaves. This, in turn, led to a rapid expansion in the merchant navies maintained by many countries. In the past, the captains of these ships relied upon local knowledge and expertise. They always plied the same route. As trade developed, this expertise became less important. As a result, there was an increasing need to produce accurate maps and navigation charts. These involved the calculation of precise distances, longitudes, as needed for navigation, for example.

The demand for navigational aids fuelled the development of computing devices. Charles Babbage (1791-1871) was a British mathematician and inventor, whose early attempts were funded by the Navy Board. As in previous centuries, his Difference Engine was designed to calculate a specific function (6th degree polynomials): a + bN + cN2 + dN3 + eN4 + fN5 + gN6.

This machine was never completed. In contrast, however, Babbage's second machine, the Analytical Engine, was a more general computer. This created the problem of how to supply the machine with its program. Punched cards were used and became perhaps the first solution to a user interface problem. The idea was so popular that this style of interaction dominated computer use for the next century.

1.2.3 The Early Twentieth Century

With the rise of mass production techniques on the east coast of the United States, economic pressures for trade increased. This had the effect of drawing migrants who were fleeing famines in both Ireland and Scandinavia. The rapid influx of people caused severe problems for the United States government. They wanted to monitor this flow in order to avoid the introduction of epidemics from certain parts of the world. They also wanted to build a profile of the population for tax reasons.

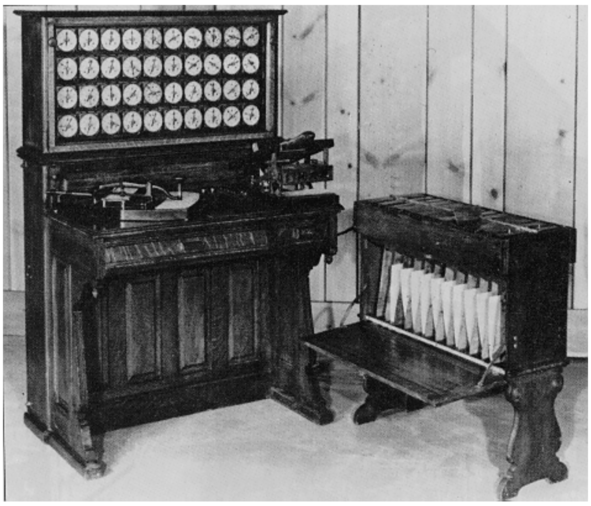


Figure 1.1 The Hollerith punch card and tabulating machine.
From http://www.columbia.edu/acis/history/

As a result, Herman Hollerith (1860-1929) was recruited by the American census office to develop a computational device to calculate general statistics for the immigrant population. In 1887, he developed a punched card tabulating machine (see Figure 1.1) that could sort more than 200 cards a minute. As a result the 1890 census took about one-third of the time of the 1880 census.

These early attempts led to the foundation of the Computer-Tabulating-Recording Company (1911). This was possibly the first and the biggest computer company. In 1914 Thomas J. Watson (Snr) joined the organization and built it into the International Business Machines Corporation (IBM).

The important point here is that economic and political factors were intervening to create a greater market for computing devices. The term 'computer' was originally used to describe the people who manually performed these calculations in the early twentieth century. In these early machines, the style of interaction was still based around the techniques pioneered in Babbage's Analytical Engine. Sequences of instructions were produced on punched cards. These were entered in batch mode, the jobs were prepared in advance and 'interaction' was minimal.

1.2.4 The Mid-Twentieth Century

The Second World War created another set of 'narrow' applications for computing devices. In particular, Alan Turing, an English logician and a founder of Computer Science, was employed to break the German encryption techniques. This led to the development of the Colossus (1943) that was perhaps the first truly interactive computer. The operator could type input through a keyboard and gain output via a teleprinter.

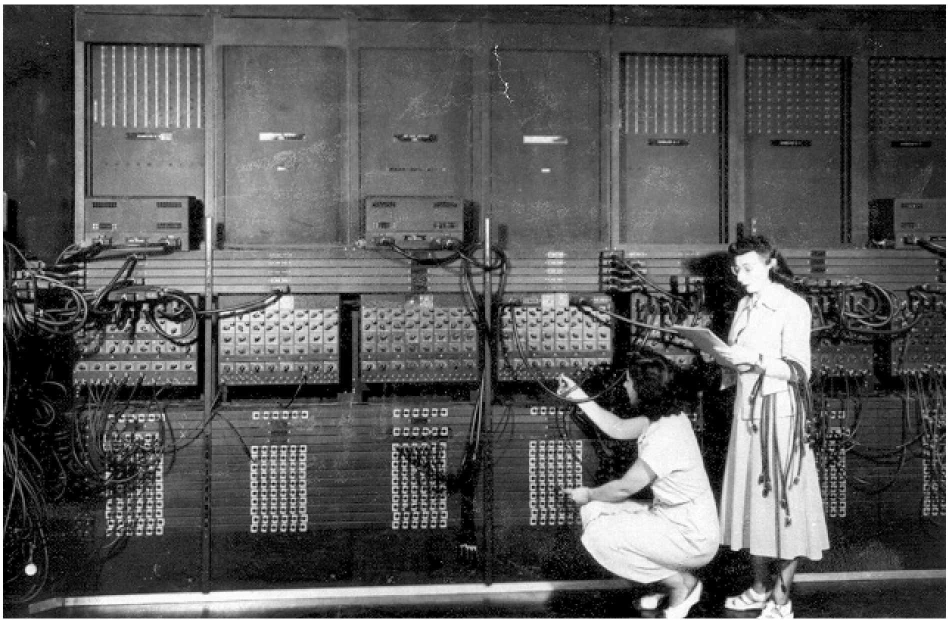


Figure 1.2 Eniac (1943)
From http://www.math.ntnu.no/~ronquist/kurs/super/2004h

Many of the Colossus techniques were also introduced in the ENIAC machine (see Figure 1.2), the first all-electronic digital computer, produced around 1946 by J. W. Mauchly and J. P. Eckert in the United States. As with Colossus, the impetus for this work came from the military. In this case they were interested in ballistic calculations. To program the machine, you had to physically manipulate 200 plugs and 100-200 relays. Figure 1.3 shows the Manchester Mark I computer from about this period.

In 1945 Vannevar Bush, an electrical engineer in the USA, published his 'As we may think' article in Atlantic Monthly. This article was the starting point of Bush's idea of the Memex system. The Memex was a device in which individuals could store all personal books, records, communications, etc., and from which items could be retrieved rapidly through indexing, keywords and cross-references. The user could annotate text with comments, construct a trail (chain of links) through the material and save it. Although the system was never implemented, and was a device based on microfilm record rather than computers, it conceived the idea of hypertext and the World Wide Web (WWW) as we know it today.

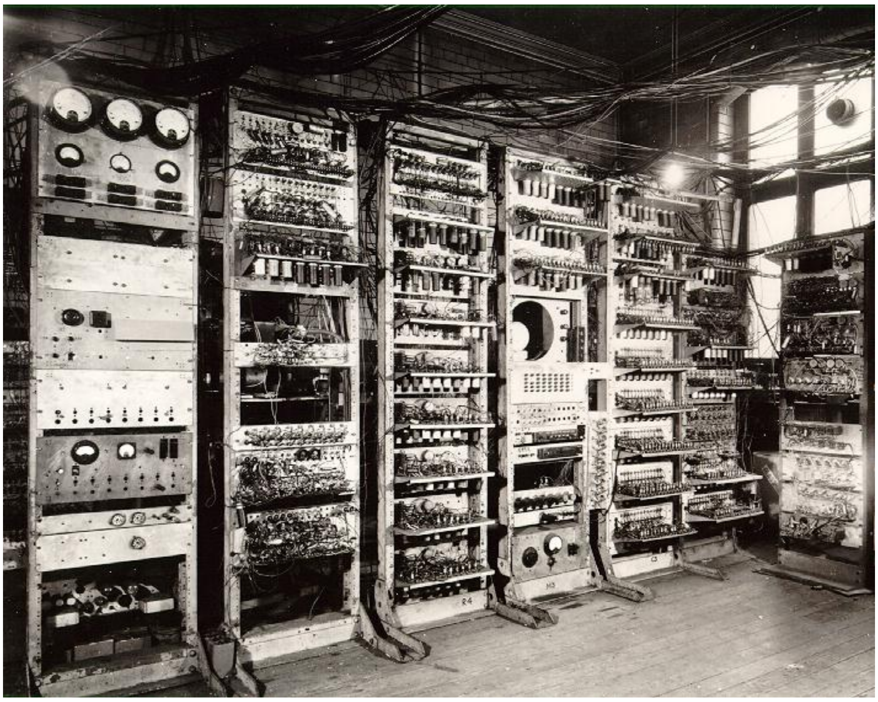


Figure 1.3 Manchester Mark I Computer
From http://to55er.files.wordpress.com/2009/10/mark1.jpg

By this time, the first machine languages were also beginning to appear. These systems were intended to hide the details of the underlying hardware from programmers. In previous approaches, one was required to understand the physical machine. In 1957 IBM launched FORTRAN, one of the first high-level programming languages which created a new class of novice users – people who wanted to learn how to program but who did not want a detailed understanding of the underlying mechanisms. FORTRAN was based on algebra, grammar, and syntax rules, and became the most widely used computer language for technical work.

In the early 1950s, some of the earliest electronic computers, such as MIT's Whirlwind and the SAGE air-defence command and control system, had displays as integral components. By the middle of the 1950s it became obvious that the computer could be used to manipulate pictures as well as numbers and text. Probably the most successful in this area was Ivan Sutherland who, in 1963, developed the SketchPad system at the MIT Lincoln Laboratory. It was a sophisticated drawing package which introduced many of the concepts found in today's interfaces, such as the manipulation of objects using a light-pen, (including grabbing objects, moving them, changing their size) and using constraints and icons.. Hardware developments that took place during the same period include 'low-cost' graphics terminals, input devices such as data tablets, and display processors capable of real-time manipulation of images.

Two of the most dominant influences in suggesting the potential of the technology of this era have been

Doug Engelbart and Ted Nelson. They both took the concept of the Memex system and elaborated on it in various ways. While Nelson focussed on links and interconnections (which he named 'hypertext' and implemented as the Xanadu system), Engelbart focussed primarily on the hierarchic structure of documents. In 1963 he published an article 'A conceptual framework for augmenting human intellect', in which he viewed the computer as an instrument for augmenting man's intellect by increasing his capability to approach complex problem situations.

1.2.5 Turning Points

The turning points in the development of computers that would allow it to become available to the man in the street occurred in the middle 1970s – also the period which saw the rise of two major American role players in today's computer industry: Microsoft and Apple Computers. Initial attempts to support the 'desktop metaphor' pushed graphical facilities and processor speeds to their limits. Below we follow the development towards where we are today.

1.2.5.1 Apple's First Personal Computer Kits (1976)

The Apple Company was founded by Steve Jobs and Steven Wozniak in 1976. Initially, they produced a series of kit machines similar to those that led to the development of the IBM PC a few years later. They hit upon the idea of pushing the code needed to represent the desktop into hardware. Graphics and device handling were burned into ROM (read only memory). This led to a higher degree of consistency because it became more difficult to change the look and feel of the interface. (The Apple history website http://www.apple-history.com/ provides greater detail on their early history.)

Apple I (Figure 1.4) was Steven Wozniak's first personal computer. It made its first public appearance in April 1976 at the Homebrew Computer Club in Palo Alto, but few took it seriously. It was sold as a kit one had to assemble. At $666, it was an expensive piece of machinery by today's standard, considering that the price only included the circuit board. Users even had to build the computer case themselves. It was based on the MOStek 6502 chip (most other kit computers used the Intel 8080 chip).

Figure 1.4 Apple I
From essentialsns.com/Services.htm

1.2.5.2 IBMs First Personal Computer (1981)

Before the 1980s, personal computers were only used by enthusiasts. They were sold in kits and were distributed through magazines and electronic shops. This meant that their user population consisted almost entirely of experts. They understood the underlying hardware and software mechanisms because they had built most of it. Many people thought that they were 'toys'. In the late seventies, this attitude began to change as the demand for low-end systems began to increase.

Figure 1.5 IBM PC
From IBM Archives (www.ibm.com/ibm/history)

In 1981, IBM introduced their first PC (see Figure 1.5) together with DOS (Disk Operating System). Little has changed in the underlying architecture of this system since its introduction. The relatively low cost and the ease with which small-scale ‘clusters’ could be built (even if they were not networked) vastly expanded the user population. A cycle commenced in which more people were introduced to computers. Increasing amounts of work were transferred to these systems and this forced yet more people to use the applications. As a result, ‘casual users’ began to appear for the first time. These are people whose work occasionally requires the use of a computer but who spend most of their working life away from a terminal. This user group found PCs hard to use. In particular, the textual language required to operate DOS was perceived to be complex and obscure.

1.2.5.3 The Xerox Star (1982)

Although the graphical user interface (GUI) had its roots in the 1950s it was not developed until the 1970s when a group at the Xerox Palo Alto Research Center (PARC) developed the Alto, a GUI-based computer. The Alto was the size of a large desk, and Xerox believed it to be unmarketable. In 1982, Xerox introduced their STAR user interface. This marks what many people believe to be the beginning of HCI as a conscious design activity by software companies. As a response to the increasing use of PCs by casual users and in office environments, Xerox began to explore more ‘intuitive’ means of presenting the files, directories and devices that were represented by obscure pieces of text in DOS. Files were represented by icons and were deleted by dragging them over a wastebasket. Other features or principles included a small set of generic commands that could be used throughout the system, a high degree of consistency and simplicity, a limited amount of user tailorability, what you see is what you get (WYSIWYG), and the promotion of recognising/pointing rather than remembering/typing. It was the first system based upon usability engineering. Ben Shneiderman at the University of Maryland coined the term ‘direct manipulation’ in 1982 and introduced the psychological foundations of computer use (Myers, 1998). In the Units to follow we will come back to the importance of some of these principles in interface design.

Steve Jobs of Apple Computers took a tour of PARC in 1979, and saw the future of personal computing in the Alto. Although much of the interface of both the Apple Lisa (1983) and the Apple MacIntosh (Mac) (1984) was based (at least intellectually) on the work done at PARC, much of the Mac OS (operating system) was written before Jobs’ visit to PARC. Many of the engineers from PARC later left to join Apple. When Jobs accused Bill Gates of Microsoft of stealing the GUI from Apple and using it in Windows 1.0, Gates fired back: ‘No, Steve, I think it’s more like we both have a rich neighbour named Xerox, and you broke in to steal the TV set, and you found out I’d been there first, and you said. ‘Hey that’s not fair! I wanted to steal the TV set!’

ACTIVITY 1.1

Complete the time line of Table 1 for the historical context of human-computer interaction by supplying the missing information.

Table 1: The historical context of human-computer interaction

< 1450	Persian astrologer, ________, used a device to calculate the conjunction of planets.
1600	The German mathematician, ________, developed a tool to perform simple addition and subtraction.
	Blaise Pascal built a simplified replica of ________'s device.
1820 - 1870	Charles Babbage built his ________ to calculate 6^{th} degree polynomials.
	________ developed his Analytical Engine, which was the first calculating device to use punched cards.
________	Herman Hollerith developed a computational device, using punched cards, to calculate census statistics.
1914	Thomas J Watson joined the ________ Company and built it up to form the International Business Machines Corporation (IBM).
1943	________ developed the ________ to try and break German encryption techniques. The Colossus accepted input via a keyboard and produced output via a teleprinter.
________	Vannevar Bush published his 'As we may think' article in Atlantic Monthly, introducing his Memex system.
1946	The ________ machine, the first all electronic digital computer, was produced by J W Mauchly and J P Eckert in the United States.
________	IBM introduced the FORTRAN high-level programming language.
1963	Ivan Sutherland developed the ________ system at the MIT Lincoln Laboratory, the first sophisticated drawing package.
1976	Steven Wozniak produced Apple I, based on the ________ Chip.
________	IBM produced their first PC with DOS.
1982	Xerox produced their ________, in which files were represented by icons and were deleted by dragging them over a wastebasket. This marked the advent of the modern desktop.

The fact that both Apple and Microsoft got the idea of the GUI from Xerox put a major dent in Apple's lawsuit against Microsoft over the GUI several years later. Although much of The Mac OS was original, it was similar enough to the old Alto GUI to make a 'look and feel' suit against Microsoft doubtful. Today the look and feel of the Microsoft Windows Environment and the Macs are very similar, although both have retained some of their original unique features and identities (also in the naming of features). As far as hardware is concerned, the Apple and the PC have developed in more or less the same direction. The only difference is that Apple has experimented beyond pure functionality as far as the aesthetics of their machines is concerned. Figures 1.6 and 1.7 show some examples.

Figure 1.6 Apple Macintosh Classic II (1991)
From www.apple-history.com

Figure 1.7 iMac 17” (2002)
From www.apple-history.com

1.2.5.4 The Internet, World Wide Web and Social Networks

Early in 1962, Rand Corporation, one of America’s leading military suppliers, became concerned about how people would communicate after a nuclear holocaust. Their solution (called ARPANET) was to grow into the Internet – a highly connected network of computer systems. Since the inception of the Internet, there has been a rapid growth in world-wide computer networks. In 1971, there were twenty-three host machines. In 1980 there were approximately one hundred computers attached to the Internet and in 1990 one hundred thousand. In 1994, the number of systems connected to the Internet exceeded one million. A 1999 estimate placed the number of Internet users at more than two hundred million, in 2002 there were over five hundred million and in 2009 there were 1.73 billion internet users worldwide.

Hundreds of sites in many different domains provide access to a vast range of information sources. The growth of these information sources and the development of applications such as Internet Explorer, Netscape, and Mosaic encouraged the active participation of new groups of users. Most of these participants possess only a minimal knowledge of the communications mechanisms that support computer networks.

Two major developments built on the Internet are the use of electronic mail systems (e-mail) and the World Wide Web (WWW), and lately, web based social networks:

- **E-mail**

Until the late 1980s the growth in electronic mail was largely restricted to academic communities, i.e. universities and colleges. It then became increasingly common for companies to develop internal mail systems, which were typically based around proprietary systems that were sold as part of a PC networking package. Most large businesses could not see the point of hooking up to the Internet and so addresses were only valid within that local area network. Concerns over Internet security also encouraged businesses to isolate their users’ accounts from the outside world. However, the situation has changed. The ability to rapidly transfer information using systems such as Microsoft's Internet Explorer and Netscape has encouraged companies to extend their e-mail access.

In 2009 close to 250 billion e-mails were being sent daily (http://royal.pingdom.com/2010/01/22/internet-2009-in-numbers/).

- **The World Wide Web (WWW)**

The WWW grew from the National Centre for Supercomputer Applications (NCSA), University of Illinois and from CERN, a European Research Centre for nuclear physics, where there were concerted efforts to improve the means of passing files over remote networks. Application or client programs, called browsers, translate user request for information into the communications primitives that are necessary to transfer relevant data from remote servers. The work at the NCSA led to the development of the Mosaic Web browser in 1993. Mosaic was a free program that did much to attract the initial user group to the Web. Netscape was then developed as a commercial successor once there were enough users for the Web to be successful. Users were reluctant to pay for a browser until there was enough information on the Web. Microsoft's Internet Explorer followed soon after.

An extensive user community has developed on the Web since its public introduction in 1991. In the early 1990s, the developers at CERN spread word of the Web's capabilities to scientific audiences worldwide. By September 1993, the share of Web traffic traversing the NSFNET Internet backbone reached 75 gigabytes per month, or one percent. By July 1994 it was one terabyte per month, and in the beginning of the 2000s in excess of ten terabytes per month. In 2009 there were more than 230 million web sites and 1.73 billion internet users worldwide. Close to 250 billion e-mails were being sent daily by 2009 (http://royal.pingdom.com/2010/01/22/internet-2009-in-numbers/).

- **Social networks**

A social network is a social structure that connects individuals (or organisations). Connections are based on concepts such as friendship, kinship, common interest, financial exchange, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige. The WWW and mobile technology have become important platforms for new forms of social networks. Probably the best known examples are Facebook and Twitter.

Facebook is a social networking website that was started in February 2004 by Mark Zuckerberg (then 20 years old). Members of Facebook create personal profiles, photo albums and information 'walls' to share happenings in their lives with people around the world. Facebook profiles are not exclusively for individuals. Schools, associations, companies, and so on can also have profiles and friends on Facebook. Anyone older than 12 can become a Facebook user and it costs nothing. In 2010 the website had more than 400 million active users worldwide.

Networks such as Facebook do not come without problems. Facebook has been banned in several countries due to its use to spread political propaganda. Many companies block their employees from accessing Facebook to prevent them from wasting their time on the network during work hours.

Twitter is a social networking and microblogging service that enables its users to communicate through 'tweets'. It was created in 2006 by Jack Dorsey. Tweets are text-based messages (maximum 140 characters) that appear on the author's profile page for viewing by people subscribed to that page (they are known as the author's 'followers'). Since late 2009, users can follow lists of authors instead of following individual authors. Users can send and receive tweets via the Twitter website, external applications or Short Message Service (SMS). Twitter had over 100 million users worldwide in 2010.

Members of both Facebook and Twitter can restrict viewing of their information to users who are registered as their friends or followers, or they can allow open access.

1.2.5.5 Mobile Computing

Mobile computation can take place over large distances using cellular and satellite telephone links. It has made internet access an integral part of everyday life through notebook computers, personal digital assistants (PDAs) like the iPhone®, and standard cell phones.

Figure 1.8 Three mobile devices from Apple:
the iPad (2010), the MacBook (late 2009), and the iPhone 4 (2010)
From www.apple-history.com

Mobile laptop and notebook computers can use one of two types of wireless access services when away from the home or office:

- WiFi uses radio waves to broadcast an Internet signal from a wireless router to the immediate surrounding area. If the wireless network is not encrypted, anyone can use it. WiFi is commonly used in public places to create 'hotspots'.
- Cellular broadband technology typically involves a cellular modem or card to connect to cell towers for Internet access. The modem or card is inserted into a notebook computer when the user wants to access the internet.

Cellular broadband connections also make it possible to provide Internet access through cell phones and PDAs. The latter depends on the model of phone and on the type of contract with the service provider.

1.3 The Current Context and Future Directions

1.3.1 The Current Context

The following aspects of computer use currently affect HCI:

- **Distributed systems**: The development of innovative user interfaces is increasing access to distributed information sources. People 'surfing the net' are no longer just programmers looking for interesting pieces of code. The term 'Web 2.0' refers to web applications that make interactive information sharing and collaboration on the World Wide Web possible. Users contribute to the content of the web site, where first generation web sites were limited to passive viewing of information. Facebook (discussed above) is an example of a Web 2.0 web site. The next generation web applications (Web 3.0) involve web-based applications that 'talk' to one another to produce new information. Content is thus generated by the web applications themselves.
- **Multimedia interfaces**: Text is still the most significant form of interaction with computer systems. Increasingly, however, we have the problem of integrating it into graphical, video, and audio information sources.
- **Advanced operating systems**: Many of the changes described in section 1.2 have been being driven by changes in the underlying computer architecture. Increasing demands are made upon processing resources by graphical and multimedia styles of interaction. These demands are being met by the improvement of operating systems such as OS2 and Windows 2007 which allows, for example, for much improved manipulation of multimedia documents.
- **HCI development environments**: On top of the new generations of operating systems, there are new generations of interface development software. Many of these environments extend the graphical interaction techniques of the Apple and Windows desktops to the construction of the interface itself. For perhaps the first time, users may be able to customise their working environment. This creates opportunities but also carries high risks if many different users must all operate the same application at different times.
- **Ubiquitous computing (UbiComp)**: This refers to computer systems that are embedded in everyday objects and thus, unobtrusively, become part of the environment. An example is a computerised control system found in a modern car (for example, activating the windshield wipers at the appropriate wiping speed when it detects rain).
- **Mobile technology**: This has changed the context within which technology is used, the compilation of the user population, as well as the design of user interfaces. Computers (in their mobile form) can be used any time, any place. Through mobile technology, people who would never have access to computers in their non-mobile form, now have mobile phones through which they can access resources such as the WWW. What could previously only be done on a desktop PC can bow be done on mobile phones or devices. In 2007, 77% of all Africans had mobile phones, while only 11% had computer access. To support this market, designers have to find ways to create user interfaces that fit into the small displays of mobile devices.

1.3.2 Future Directions

Mobile and ubiquitous computing will remain focus areas of the future. Harper, Rodden, Rogers, Sellen (2008) have identified five major transformations in computing, that will affect the field of HCI increasingly in the next decade. These are:

1. **The changing notion of 'the interface'.**
   Old ideas of what an interface is will not apply in the future era of ubiquitous computing. Graphical user interfaces and the mouse will increasingly be replaced by tangible interfaces controlled through touch, speech and gesture and input mechanisms using eye movement and brain activity will become more commonplace. Embedded devices have no explicit interface, which places interface design in a completely different perspective.

2. **Increasing dependency on technology.**
   For older people who have grown up without the Internet and mobile phones it is still possible to imagine life without them. However, younger generations who have always had these instantly available will not be able to function without them. Users are thus constantly becoming more knowledgeable about the use of technology, but they are also losing skills that older generations had because they did not have constant access to technology (e.g. mental calculation skills, and possibly memory skills).

3. **Hyper-connectivity.**
   Communication technology will continue to improve and allow even more forms of connectivity among people. The rapid growth in connectivity will impact on the way we relate to people, how we make friends and how we maintain relationships. The etiquette of when, how, and with whom we communicate, is also changing. For example, students send e-mail to their lecturers using the same slang they would use with their friends, while, in person, they would never speak that way to the lecturer. People engage in romantic relationships with people whom they have never met face-to-face. Where previously there was a clear distinction between work space (or time) and leisure space (or time), the levels of connectivity have blurred these boundaries. The question is now: What will the effect of this be on our social make-up in the long run?

4. **Changes in the means of and reasons for recording information.**
   Things that were previously only stored in people's memories are now being recorded in digital format. Mobile phones with cameras and video recording capabilities allow people to record activities that would previously be forgotten. The replacement of handwritten letters with e-mails means that it is much easier to build archives of everyday communications. The increase in recorded information requires improved systems for managing storage and access of the information.

5. **Increased creativity through technology.**
   Increasingly accessible and flexible computing devices can support new ways of playing, learning and creating. They become tools that can augment human cognition by visualising complex data or processes or by processing huge amounts of information in short periods of time.

ACTIVITY 1.2

Name and describe one specific example of each of the following:

- A mobile device
- A web-based social network (excluding Facebook and Twitter)
- A ubiquitous computing device.

1.4 HCI and Related Concepts and Fields

HCI emerged in the early 1980s as a specialty area in computer science and has since then developed as an area of research and practice that attracts professionals from a wide range of disciplines (Carroll, 2009).

1.4.1 Definitions of HCI

Various definitions for the term 'human-computer interaction' have been put forward over the years. Here are a few:

- HCI is a 'set of processes, dialogues, and actions through which a human user employs and interacts with a computer' (Baecker and Buxton, 1987).
- HCI is a 'discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. From a Computer Science perspective, the focus is on interaction and specifically on interaction between one or more humans and one or more computational machines' (ACM SIGCHI, n.d.). A computational machine (computer) is defined to include traditional workstations as well as embedded computational devices, such as spacecraft cockpits or microwave ovens, or specialized boxes such as electronic games. A human is defined to include a range from children to the elderly, computer aficionados to computer despisers, frequent users to hesitant users, big hulking teenagers to people with special needs, etc.
- HCI is 'the study of people, computer technology, and the ways these influence each other' (Dix et al., 2004). A (human) user is defined as whoever is trying to accomplish something using the technology and can mean an individual user, a group of users working together, or a sequence of users in an organization, each dealing with some part of the task or process. The computer is defined as any technology ranging from the general desktop computer to large-scale computer systems, a process control system, or an embedded system. The system may include non-computerized parts, including other people. Interaction is defined as any communication between a user and the computer, be it direct or indirect. Direct interaction involves a dialogue with feedback and control during performance of the task. Indirect interaction may involve background or batch processing.
- HCI is concerned with studying and improving the many factors that influence the effectiveness and efficiency of computer use. It combines techniques from psychology, sociology, physiology, engineering, computer science, and linguistics (Johnson, 1997).

There are several other terms and fields of study that have a strong connection with HCI. Some are listed below:

- *Ergonomics* is the study of work. The term is most widely used in the United Kingdom and Europe, in contrast to the United States and the Pacific basin where the term 'Human Factors' is more popular (see below). Ergonomics has traditionally involved the design of the 'total working environment'; this includes the height of the chair, table, etc. Health and safety legislation, such as the UK Display Screen Equipment Regulations (1992), is increasingly blurring the distinction between HCI and ergonomics. In order to design effective user interfaces, we must consider wider working practices. For instance, the design of a telesales system must consider the interaction between the computer application, the telephone equipment and any additional paper documentation.
- *A human factor is* a term used to describe the study of user interfaces in their working context. It addresses the 'entire person' and includes:
    - o Physiology, our physical characteristics such as height and reach.
    - o Perception, our ability to sense information, hearing, touch, and sight.
    - o Cognition, the way we process data, such as the information we extract from a display.

  It has much in common with ergonomics but is often used to refer to HCI in the context of safety-critical applications. Physiological problems, etc., have a greater potential for disaster in these systems.
- *Usability* is defined by the International Standards Organization (ISO) as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". The ISO standard 9241 gives the following definition of the components:
    - o Effectiveness: The accuracy and completeness with which specified users can achieve specified goals in particular environments.
    - o Efficiency: The resources expended in relation to the accuracy and completeness of goals achieved.
    - o Satisfaction: The comfort and acceptability of the work system to its users and other people affected by its use.
- *User experience* refers to how people feel about a product. How satisfied are they when using it, looking at it, or handling it? It includes the overall impression as well as the small effects such as how good it feels to touch. According to Preece et al. (2007) you cannot design a user experience, you can only design *for* user experience.
- *Interaction design* is defined by Preece et al. (2007, p. xvii) as "designing interactive products to support the way people communicate and interact in their everyday and working lives". It involves four activities, namely:
    - o Identifying needs and establishing user requirements.
    - o Developing alternative designs according to the requirements.
    - o Building prototypes of the designs so that they can be assessed.
    - o Evaluating the designs and the user experience.
- *Accessibility* in the context of HCI is 'designing products so that people with disabilities can use them. Accessibility makes user interfaces perceivable, operable, and understandable by people with a wide range of abilities, and people in a wide range of circumstances, environments, and conditions. Thus accessibility also benefits people without disabilities, and organizations that develop accessible products' (Henry, 2007). It is the degree to which a system is usable by people with disabilities (Preece at al., 2007). Some people see accessibility as a subset of usability, while others regard it is a prerequisite for usability.
-

ACTIVITY 1.3

Define the following terms:

- Human-computer interaction
- Ergonomics
- Usability
- Interaction design
- Accessibility

1.4.2 Who is Involved in HCI?

HCI is a multidisciplinary or interdisciplinary subject. The ideal designer of interactive systems should have expertise in a variety of topics (Dix et al., 2004, Preece et al., 2007), including:

- Psychology and cognitive science, giving insight into the user's capabilities, and perceptual, cognitive, and problem-solving skills.
- Environmental factors and ergonomics to be able to address the user's working environment, physical capabilities and comfort factors.
- Organizational factors to be able to address training, job design, productivity, and work organization.
- Health and safety factors.
- Philosophy, sociology, and anthropology to help understand the wider context of interaction.
- Linguistics.
- Computer science and engineering to be able to build the necessary technology.
- Graphics design to produce effective interface presentation.

There is too much expertise here to be held by one person, even by the average design team. In practice, designers tend to take a strong stance on one side or another. It is, however, not possible to design effective interactive systems from one discipline in isolation. There is a definite interaction among all these disciplines in designing and developing an interactive artefact.

Professionals in HCI are widely spread across the spectrum of subfields. They are, for example, user experience designers, interaction designers, user interface designers, application designers, usability engineers, user interface developers, application developers, or online information designers (Carroll, 2009).

ACTIVITY 1.4

Do your own research on the Internet to find out what each of the occupations below entail. Then, assuming you are the CEO of a dynamic new web application development company, formulate job advertisements for each of the positions. Include the required qualifications and experience as well as the key tasks that the person will perform.

- Usability engineer
- Interaction designer
- User experience designer.

1.5 Conclusion

The purpose of studying human-computer interaction is to improve the quality of interaction between human and machine by systematically applying knowledge about human capabilities and limitations, and machine capabilities and limitations; also, to improve the productivity, functionality, effectiveness, efficiency, and usability of technology. Human-computer interaction is about designing for people (users). It is, however, not only about the display and the keyboard, or about people in offices. It has a much wider context, addressing interactive situations in everyday life as well. Therefore, although the main purpose of this module is to introduce you to HCI, the aim is also to create an awareness of user-centred design in general. User-centred design is not easy but will increase in importance with the changes taking place in technology.

NOTES

UNIT 2
Human Issues in HCI

CONTENTS

UNIT 2 - OUTCOMES

After studying this unit you should:

- Understand how perception and cognition influence user interface design.
- Understand the differences between short and long-term memory and their influence on interface design.
- Know how physical attributes of users can affect their interaction with computers.
- Understand how users may differ in terms of culture, personality, gender and age, and how this impacts on HCI.
- Understand how different levels of expertise influence users' interaction with a system.
- Know the kinds of errors humans make, why they make them and how this should be addressed in user interface design.

2.1 Introduction

In this unit we focus on the 'human' in human-computer interaction. We address some of the differences in and between user populations that must be considered when developing and installing computer systems. In particular, we will identify the effects that perception, cognition, and physiology can have upon human performance. We will also touch on the issues of personality and cultural diversity and discuss the special needs and characteristics of users in different age groups.

Part of human nature is to make errors. We look at the different kinds of errors people make and discuss ways to avoid them.

Not all of this information will be relevant to all commercial applications. For instance, the developers of a mass market database system may have little or no control over the workstation layout of their users. In other contexts, particularly if you are asked to install equipment within your own organisation, these factors are under your personal control. It is, however, important that, as future software designers and Information Technology managers, you are made aware of the factors that influence people's experience with technology.

2.2 Cognitive Psychology in HCI

Many cognitive processes underlie the performance of a task or action by a human. Human information processing consists of three interacting systems: the perceptual system, the cognitive system, and the motor system. We can therefore characterise human (user) resources into three categories:

- Perception: the way that they detect information in their environment.
- Cognition: the way that they process that information.
- Physiology: the way in which they move and interact with physical objects in their environment.

A vital foundation for designers of interactive systems is an understanding of the cognitive and perceptual abilities of the user. Some regard perception as part of cognition (Preece et al., 2007) but here we will discuss it as a separate aspect of human information processing.

2.2.1 Perception

Perception involves the use of our senses to detect information. The human ability to interpret sensory input rapidly and initiate complex actions makes the use of modern computer systems possible. In computerised systems this mainly involves using senses to detect audio instructions and output, visual displays and output, and tactile (touchable) feedback.

Information from the external world is initially registered by the modality-specific sensory stores or memories for visual, audio, and tactile information respectively. These stores can be regarded as input buffers holding a direct representation of sensory information, but the information persists there for only a few tenths of a second. So, if a person doesn't act on sensory input immediately it will not have an effect.

Many factors affect perception, for example:

- Change in output, such as changes in the loudness of audio feedback or in the size of elements of the display.
- Maximum and minimum detectable levels of, for example, sound. Different people can hear different frequencies. People also differ in the number of different signals they can process at a time.
- The field of perception. Depending on the environment, not all stimuli may be detectable. All parts of the display may not be visible to a user if facing it at the wrong angle, for example.
- Fatigue and circadian (biological) rhythms. When people are tired their reactions to stimuli may be slower.
- Background noise.

Designers have to make sure that people can see or hear displays if they are to use them. In some environments this is particularly important. For instance, most aircraft produce over 15 audible warnings. It is relatively easy to confuse them under stress and with high levels of background noise. Such observations may be worrying for the air traveller, but they also have significance for more general HCI design. We must ensure that signals are redundant (i.e. it must be ‘more’ than are needed, desired, or required). If we display critical information through small changes to the screen, then many people will not detect the change. If you rely upon audio signals to inform users about critical events then you exclude people with hearing problems or people who work in a noisy environment. On the other hand, audio signals may irritate users in shared offices.

Partial sight, ageing and congenital colour deficits all produce changes in perception that reduce the visual effectiveness of certain colour combinations. Two colours that contrast sharply when perceived by someone with normal vision may be far less distinguishable to someone with a visual disorder. People with colour deficits generally see less contrast between colours than someone with normal vision. Lightening light colours and darkening dark colours will increase the visual accessibility of a design.

Three aspects of colour influence how they are perceived:

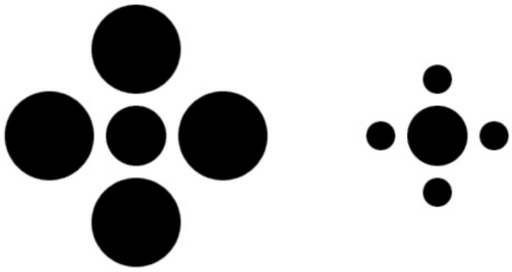
- Colour *hue* describes the perceptual attributes associated with elementary colour names. Hue enables us to identify basic colours, such as blue, green, yellow, red and purple. People with normal colour vision report that hues follow a natural sequence based on their similarity to one another.
- Colour *lightness* corresponds to how much light appears to be reflected from a surface in relation to nearby surfaces. Lightness, like hue, is a perceptual attribute that cannot be computed from physical measurements alone. It is the most important attribute in making contrast more effective.
- Colour *saturation* indicates a colour’s perceptual difference from a white, black or grey of equal lightness. Slate blue is an example of a desaturated colour because it is similar to grey.

Congenital and acquired colour deficits typically make it difficult to discriminate between colours on the basis of hue, lightness or saturation. Designers can help to compensate for these deficits by using colours that differ more noticeably with respect to all three attributes.

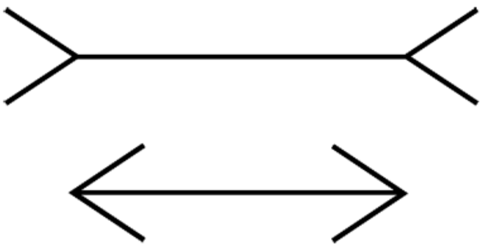
ACTIVITY 2.1

Look at the following images and answer the questions associated with each.

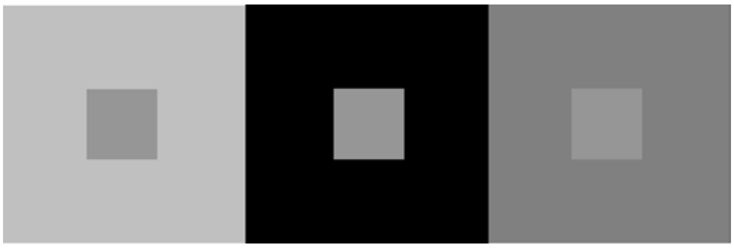
Which of the two circles in the middle are the biggest?



Which of the parallel lines are the longest?



Which of the squares in the middle is the darkest/lightest in colour?



If you use a ruler in 1 and 2 you will get the correct answer. 1 is the Ebbinghaus illusion and 2 the Müller-Lyer illusion. In 3 there is actually no difference in the colour of the three inner squares. These examples show just how powerful external influences can be in our perception of things. You can also see that this is a good font to use if you want the reader to struggle to decipher it.

2.2.2 Cognition

Cognition refers to a variety of processes that take place in our heads. These include:

- Short-term memory and information processing
- Long-term memory and learning

- Problem solving
- Decision making
- Attention
- Search and scanning
- Time perception.

Knowledge of these will help designers to create usable interfaces. Our discussion of cognition will be limited to attention and memory.

2.2.2.1 Attention

Attention is the process of concentrating on something (e.g. an object, a task or a conversation) at a specific point in time. It can involve our senses like looking at the road while driving or listening to a news story on the radio, or it can involve thinking processes such as concentrating on solving a mathematical problem in your head. People differ in terms of their attention span. Some people's attention can be distracted easily while others can concentrate on a task irrespective of external disturbances.

Attention is influenced by the way information is presented as well as by people's goals (Preece et al., 2007). This has implications for designers of computer systems. If information on an interface is poorly structured, users will have difficulty in finding specific information. How information is displayed determines how well people will be able to perform a searching task. When using a system with a particular goal in mind, the user's attention will remain focussed more easily than when he or she is aimlessly browsing through an application. Designers of browsing and searching software should therefore find ways to lead users to the information they want in a clear and focused way. In a computer game it is also important that users always know what their next goal in the game is, otherwise they will lose interest.

2.2.2.2 Memory

Memory consists of a number of systems that can be distinguished in terms of their cognitive structure as well as on their respective roles in the cognitive process (Gathercole, 2002). Different authors have different views on how memory is structured, but most distinguish between long term and short term memory. Short term memory (STM) store information or events from the immediate past and retrieval is measured in seconds or sometimes minutes (Gathercole, 2002). Long term memory (LTM) holds information about events that happened hours, days, months or years ago and the information is usually incomplete.

STM has a relatively short retention period and is limited in the amount of information that it can keep. It is easy to retrieve information from STM. Some people refer to STM as 'working memory', since it acts as a temporary memory that is necessary to perform our everyday activities. Effectiveness of STM is influenced by attention – any distraction can cause information to vanish from STM. Generally, people can keep up to seven items (e.g. a seven digit telephone number) in STM unless there is some distraction.

LTM, in contrast, has high capacity. As its name suggests, it can store information over much longer periods of time, but access is much slower. It also takes time to record memories there. If we have to extract the information from LTM it may involve several moments' thought: for example, naming the seven dwarfs or the current members of the national soccer team. The information stored in LTM is affected by people's interpretation of the associated events or context. Information retrieved from LTM is also influenced by the retriever's current context or state of mind.

We should design interfaces that make efficient use of short-term memory. Users should only be required to keep a few items of information in STM at any point during interaction. They should not be compelled to search back through dim and distant memories of training programs in order to operate the system. User interfaces can support short-term memory by including cues on the display. This is effectively what a menu does: it provides fast access to a list of commands that do not have to be remembered. In contrast, help facilities are more like long-term memory. We have to retrieve them and search through them to find the information that we need.

In line with general STM capacity, seven is often regarded as the 'magic number' in HCI. Important information is kept within the seven-item boundary. Additional information can be held, but only if users employ techniques such as chunking. This involves the grouping of information into meaningful sections. National telephone numbers are usually divided in this way, (012) 429 6122. Chunking can be applied to menus through separator lines or cascading menus.

As we have mentioned, it takes effort to hold things in STM. We all experience a sense of relief when it is freed up. As a result of the strain of maintaining STM, users often hurry to finish some tasks. They want to experience the sense of relief when they achieve their objective. This haste can lead to error. Some automatic teller machines (ATMs) suffer from this problem. Users experience a sense of closure when they have satisfied their objective of withdrawing money. Some ATMs issue the money before returning the user's card. The user then walks away and leaves his or her card in the machine. To avoid this, most ATMs do not dispense cash until the user has removed the card. The computerised system is thus designed so as to prevent error caused by limitations of STM.

An important aim for user interface design is to reduce the load on STM. We can do this by placing information 'in the world' instead of expecting users to have it 'in the head' (Norman, 1999). In computer use, knowledge in the world is provided through the use of prompts on the display and the provision of paper documentation.

ACTIVITY 2.2

1. Draw up a table with two columns – one for STM and one for LTM – and list all the differences between the two types of memory.
2. Describe your own example of how the load on the user's STM can be can be relieved through thoughtful design of the interface.

2.2.2.3 Knowledge in the World vs. Knowledge in the Head

Norman (Norman, 1999) refers to information kept in memory as 'knowledge in the head', while external information is 'knowledge in the world'. Both these kinds of information are necessary for our functioning in the world (and also for our interaction with computers), but how it is used depends on the individual. Some people rely more on knowledge in the world (e.g. notes, lists, birthday calendars), while others depend more on the knowledge in their heads (their memory). There are advantages and disadvantages to both approaches. These are summarised in Table 2.1.

Table 2.1 Comparison of knowledge in the head and the world (from Norman (1999))

Property	Knowledge in the world	Knowledge in the head
Retrievability	Easily retrievable whenever visible or audible. (Depends on availability in the environment.)	More difficult to retrieve. Requires memory search or reminding.
Learning	Learning is not required, only interpretation.	To get information there requires learning, which can be considerable.
Efficiency of use	Tends to be slowed up by the need to find and interpret the external sources.	Can be very efficient.
Ease of use at first encounter	High	Low
Aesthetics	Can be unaesthetic and inelegant, especially if there is a need to maintain a lot of information. Can lead to clutter. Requires a skilled designer.	Nothing needs to be visible, which gives the designer more freedom.

When designing interfaces, the trade-off between knowledge in the world and that in the head must be kept in mind. Do not rely too much on the user's memory, but don't clutter the interface with memory cues or information that is not really necessary. Meaningful icons and menus can be used to relieve strain on memory, but the Help menu should provide additional information 'in the world' that is difficult to display properly on the interface.

ACTIVITY 2.3

Explain how we use cellular phones as knowledge in the world. Your answer should make it clear what is meant by knowledge in the world.

2.2.2.4 Examples to Illustrate the Role of Memory in HCI

We end this section on cognition with two examples of how interface design can relate to human cognition (the user's memory in particular).

The image given in Figure 2.1 is of a message from Microsoft's Word 97. The message appeared when you spell-checked a document that contains text that you have indicated should be excluded from spell-checking (the 'no proofing' option). The message is certainly informative, but requires that the user either has an exceptional short term memory, or has pen and paper handy to write down the steps that it refers to.

Figure 2.1 Proofing in Microsoft Word (Isys, 2000)

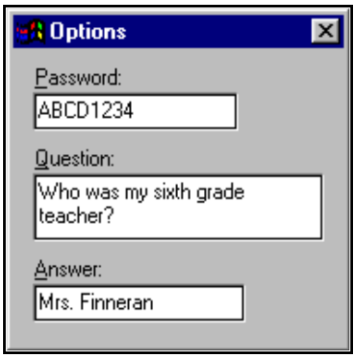


Figure 2.2 Mechanism to retrieve a forgotton password (Isys, 2000)

Given all the passwords each of us must keep track of, it's all too easy to forget the password for a particular account or program. Figure 2.2 shows how many applications nowadays help the user to remember a password. When creating a new account you are asked to specify the new password, and in addition, provide a question and answer, in the event that you forget your password at some later time. The log-in window includes an 'Forgot my password' button that will prompt you with the question you provided at registration, and await your response.

This is a good solution to a problem that has plagued system operators everywhere. It is an interface feature that should be considered for every application that requires a password.

2.3 Physiology

Physiology involves the study of the human anatomy. It might seem strange to include this in a course on user interface design, but knowledge of physiology can make a noticeable contribution to the design of a successful system.

2.3.1 Physical Interaction and the Environment

When using a computer system, as a minimum requirement users must be able to view (or perceive) the interface and reach the input devices. Designers often have relatively little influence on the working environments of their users. If they do have some power, here are a few guidelines they can follow:

- Visual display should always be positioned with the correct visual angle to the user. Even relatively short periods of rotation on the neck can lead to long periods of pain in the shoulders and lower back.
- Keyboard and mouse use: Prolonged periods of data entry place heavy stress upon the wrist and upper arm. A large range of low-cost wrist support is now available. They are a lot cheaper than the expense of employing and re-training new members of staff. Problems in this regard include repetitive strain injury and carpal tunnel syndrome (both cause pain or numbness in the arms). Frequent breaks can help to reduce the likelihood of these conditions.

- Chairs and office furniture: It's no good providing a really good user interface if your employees spend most of their time with a chiropractor. It is worth investing in well designed chairs that provide proper lower back support and promotes a good posture at the computer.
- Placement of work materials: Finally, it is important that users are able to operate their system in conjunction with other sources of information and documentation. Repeated gaze transfers lead to neck and back problems. Paper and book stands can reduce this.
- Other people: You cannot rely upon system operators to prevent bad things from happening. Unexpected events in the environment can create the potential for disaster. For example, a patient-monitoring system should not rely on a touch screen if doctors or other staff that move around the patient can accidentally brush against it.

It also pays to consider the possible sources of distraction in the working environment:

- Noise: Distraction can be caused by the sounds of other workers (phone calls, sound from their computers) and office equipment (fans, printers). There are a number of low-cost solutions, for example you may introduce screens around desks or cover devices such as printers. Higher cost solutions involve the use of white noise to mask intermittent beeps.
- Light: Bright lighting can cause distraction in interaction with computers. Its impact can be reduced by blinds and artificial lighting to reduce light in the room. A side-effect of this is that, over time, users may suffer from fatigue and drowsiness. Many Japanese firms have invested in high-intensity lighting systems to avoid this problem. Lower cost solutions involve moving furniture or using polarising filters.

There are also a number of urban myths (untruths) about the impact of computer systems on human physiology:

- Eyesight: Computer use does not damage your eyes or eyesight. It may, however, make you aware of existing defects.
- Epilepsy: Computer use does not appear to induce epileptic attacks. Television may trigger photosensitive epilepsy but the visual display units of computers do not seem to have the same effect. The effect of multimedia video systems upon this illness is still unclear.
- Radiation: The National Radiological Protection Board in the UK states that VDU's do not 'significantly' increase the risk of radiation-related illnesses.

Interfaces often reflect the assumptions that their designers make about the physiological characteristics of their users. Buttons are designed so that an 'average' user can easily select them with a mouse, touchpad or a tracker-ball. Unfortunately, there is no such thing as an average user. Some users have the physiological capacity to make fine-grained selections but others do not. Although users may have the physical ability to use these interfaces, workplace pressures may reduce their physiological ability.

A rule of thumb is: do not make interface objects so small that they cannot be selected by a user in a hurry; also do not make disastrous options so easy to select that they can be started by accident.

ACTIVITY 2.4

Choose any computer-based activity you sometimes perform, (such as selecting and playing a song, writing and sending an e-mail, or submitting an assignment through myUnisa).

Name the activity. Now mention three broad categories of human resources we use in processing an action. Relate each category to how you would, in practice, use that resource in your chosen activity.

2.3.2 Users with Disabilities

Preece et al. (2007) define accessibility as 'the degree to which an interactive product is usable by people with disabilities' (p.483). There is a large range of disabilities, including severe physical conditions such as blindness, deafness and paralysis, and less severe ones such as dyslexia and colour blindness. Then there are mental disabilities such as Down syndrome, autism and dementia. In the United State more than forty eight million people were disabled in 2006 (Kraus et al., 2006). The 2001 South African census revealed that more than two million people had some form of disability (Lehohla, 2005). It is estimated that, in 2006, more than five hundred million people around the world were disabled (United Nations, 2006).

The statistics above provide ample reason to coerce designers to take accessibility into consideration. Such considerations will have a profound impact upon the development of the user interface if these groups are within the target market. Henry (2002) lists more reasons for designing systems that are accessible to people with disabilities. These include:

- Compliance with regulatory and legal requirements: In many European countries and Australia there is a statutory obligation to provide access for blind users when designing computer systems. In 1999, an Australian blind user successfully sued the Sydney Organizing Committee for the Olympic Games under the Australian Disability Discrimination Act (DDA) due to his inability to order game tickets using Braille technology (Waddell, 2002). The United States government's Section 508 of the Rehabilitation Act also stipulates that all federal electronic information should be accessible to people with auditory, visual and mobility impairments.
- Exposure to more people: Disabled people and the elderly have good reason to use new technologies. People who are unable to drive or walk and those with mobility impairments, can benefit from accessible online shopping. Communication technologies like e-mail and mobile technology can provide them with social interaction they would otherwise not have.
- Better design and implementation: Incorporating accessibility into design results in an overall better system. Making systems accessible to the disabled will also enhance usability for users without disability.
- Cost savings: The initial cost of incorporating accessibility features into a design is high, but an accessible e-commerce site will result in more sales because more people will be able to access the site. Addressing accessibility issues will also reduce legal expenses that could result from lawsuits by users

who might want to enforce their right to equal treatment.

Advances in computer technology and the flexibility of computer software make it possible for designers to provide special services to users with disabilities. Below we consider two user groups with physical disabilities – vision and motor impairments – highlighting the limitations of normal input and output devices for them.

2.3.2.1 Users with Vision Impairments

Visually impaired people mostly experience difficulties with output display, although the mouse and other input devices also pose problems. Text-to-speech conversion can help blind users to receive electronic mail or read text files, and speech-recognition devices allow voice-controlled operation of some applications. Enlarging portions of a display or converting displays to Braille or voice output can be done with hardware and software that is easily obtainable.

Reading and navigating text or objects on a computer screen is a very different experience for a user that cannot see properly. The introduction of graphical user interfaces (GUIs) was a setback for vision-impaired users, but technology innovations such as screen readers facilitate conversion of graphical information into non-visual modes. Screen readers are software applications that extract textual information from the computer’s video memory and send it to a speech synthesizer that describes the elements of the display to the user (including icons, menus, punctuation and controls). Not being able to skim an entire page, the user has to navigate without any visual clues such as colour contrast, font or position (Phipps, Sutherland, and Seale 2002). Pages that are split into columns, frames or boxes cannot be translated accurately by screen readers.

Using the mouse requires constant hand-eye coordination and reaction to visual feedback. This complicates matters for the visually impaired. They need to execute clicking and selecting functions by means of dedicated keys on a keyboard or through a special mouse that provides tactile feedback. Users with partial sight should be allowed to change the size, shape and colour of the onscreen mouse cursor, and auditory or tactile feedback of actions will be helpful.

With regard to keyboard use, visually impaired users require keys with large lettering, high contrast between text and background, and even audible feedback when keys are pressed. Blind users usually access all commands and options from the keyboard, therefore function and control keys need to be marked with Braille or tactile identification.

2.3.2.2 People with Motor Impairments

A significant proportion of the population have motor disabilities acquired at birth or through an accident or illness. Users with severe motor impairments are often excluded from using standard devices. Low cost modifications can easily increase access and without much effort. For those confined to bed, computers, and the Internet in particular, provide a particularly satisfying and stimulating means of interaction – giving them access to resources, people and places that they would not otherwise have access to.

Users with physical impairments may have difficulty grasping and moving a standard mouse, and may find fine motor coordination and selection of small on screen targets difficult, if not impossible. Clicking, double clicking, and drag-and-drop operations pose problems for these users. Designers must find ways to make this easier, for example letting the mouse vibrate if the cursor is over the target or implementing 'gravity fields' around objects so that when the cursor comes into that field it is drawn towards the target. Another solution is provided through trackballs that allow users to move the cursor using only the thumb. Severely physically impaired users may only be able to move their heads, therefore either head operated or eye tracking devices are required to translate control onscreen cursor movements. Speech input is another alternative, but there are still high error rates (especially if the user's speech is also affected by the impairment) and it can only be used in a quiet environment.

Keyboards need to be detachable so that they can be positioned according to the user's needs and there must be adequate grip between the keyboard and desktop so that the user cannot accidentally move the keyboard around. Individual keys should be separated by sufficient space and should not require much force to press. Oversized keyboards, key guards to guide fingers onto keys and software enabled sticky keys are possible solutions for users who experience uncertain touch. Some users prefer mouse sticks or hand splints to hit buttons or designers adapt the interface so that everything is controlled with a single button.

ACTIVITY 2.5

Stephen Hawking is a well-known physicist who has written influential books such as 'A Brief History in Time'. Find information on him on the internet and then describe:

- the nature of his disability,
- how it has affected his life,
- how technology has helped him, and
- the mechanisms he uses to interact with technology.

2.4 Culture

Another perspective on individual differences has to do with cultural, ethnic, racial, or linguistic background. It seems obvious that users who were raised learning to read Japanese or Chinese will scan a screen differently from users who were raised to read English or Afrikaans. Users from cultures that have a more reflective style or respect for ancestral traditions may prefer interfaces different from those chosen by users from cultures that are more action-oriented or novelty-based.

The term 'culture' is often wrongly associated with national boundaries. Culture should rather be defined as the behaviour typical of a group or class of people. Culture is conceptualised as a system of meaning that underlies routine and behaviour in everyday working life. It includes race and ethnicity as well as other variables and is manifested in customary behaviours, assumptions and values, patterns of thinking and communicative style.

Nisbett (2003) compared the thought patterns of East Asians and Westerners classifying them as holistic and analytic respectively. Holistically-minded people tend to perceive a situation globally while analytically-minded people have a tendency to perceive an object separately from the context and tend to assign objects to categories. Based on this distinction, Yong and Lee (2008) compared how these two groups viewed a web page. They found clear differences – for example, holistically-minded people scan the whole page in a non-linear fashion, while analytically-minded people tend to employ a sequential reading pattern.

As software producers expand their markets by introducing their products in other countries, they face a host of new interface considerations. The influence of culture on computer use is constantly being researched, but there are two well-known approaches that designers follow when called on to create designs that span language or culture groups:

- *Internationalisation* that refers to a single design that is appropriate for use worldwide, among groups of nations. This is an important concept for designers of web based applications that can be accessed from anywhere in the world, by absolutely anybody.
- *Localisation*, on the other hand, involves the design of versions of a product for a specific group or community, with a unified language and culture. The simplest problem here is the accurate translation of products into the target language. For example, all text (instructions, help, error messages, labels) might be stored in files, so that versions in other languages could be generated with no or little programming. Hardware concerns include character sets, keyboards and special input devices. Other problems include sensitivity to cultural issues, such as the use of images and colour.

User interface design concerns for internationalisation are many and full of pitfalls. Whereas early designs were often excused for cultural and linguistic slips, the current highly competitive atmosphere means that more effective localization will often produce a strong advantage. Simonite (2010) reports on the online translation services that now makes it possible to have web content immediately translated into different languages. These services use a technique called statistical machine translation that is based on statistical comparison of previously translated documents. From this it creates rules for use in future translation. Google's translate services can already (in 2010) translate between 52 different languages, but the translations still contain errors and need some human intervention (Simonite, 2010).

There are many factors that need to be addressed before a software package can be internationalised or localised. These can be categorised as overt and covert factors:

- *Overt factors* are tangible, straightforward and publicly observable elements. The overt factors include dates, calendars, weekends, day turnovers, time, telephone number and address formats, character sets, collating order sequence, reading and writing direction, punctuation, translation, units of measures and currency.
- *Covert factors* deal with the elements that are intangible and depend on culture or 'special knowledge'. Symbols, colours, functionality, sound, metaphors and mental models are covert factors. Much of the literature on internationalising software has advised caution in addressing covert factors such as using metaphors and graphics. This advice should be heeded to avoid misinterpretation of the meaning intended by the developers or inadvertently offending the target culture.

An example of misinterpretation is the use of the 'trash can' icon in the Apple Macintosh user interface. People from Thailand might not recognise the American 'trash can', because, in Thailand the 'trash can' is actually a wicker basket. Some visuals are recognisable in another culture but they convey a totally different meaning. In the United States, the owl is a symbol of knowledge but in Central America, the owl is a symbol of witchcraft and black magic. A black cat is considered bad luck in the US but good luck in the UK. Similarly, certain colours hold different connotations in different cultures.

One culture may find certain covert elements inoffensive but another may find the same elements offensive. In most English-speaking countries, images of the ring or OK hand gesture may be understandable, but in France the same gesture means ‘zero’, ‘nothing’ or ‘worthless’. In some Mediterranean countries, the gesture implies a man is a homosexual. Covert factors will only work if the message intended in those covert factors is understood in the target culture. Before any software with covert factors is used, the software developers need to ensure that the correct information is communicated by validating these factors with the users in the target cultures.

2.5 Personality and Gender

Some people dislike computers or are anxious about using them; others are attracted to or are eager to use any new kind of technology. Often, members of these divergent groups disapprove or are suspicious of members of the other community. Even people who enjoy using computers may have different preferences regarding interaction styles, pace of interaction, graphics versus tabular presentations, dense versus sparse data presentation, step-by-step work versus all-at-once work, etc. These differences are important. A clear understanding of personality and cognitive styles can be helpful in designing systems for a specific community of users.

Despite fundamental difference between men and women, clear patterns of preferences in interaction have been documented. Social network sites such as Facebook and Twitter tend to have more female subscribers. Huff and Cooper (1987), in their study on sex bias in educational software, found a bias when they asked teachers to design educational games for boys or girls. The designers created game-like challenges when they expected boys as their users, and more conversational dialogs when they expected girls as users. When told to design for students, the designers produced ‘boy-style’ games.

It is often pointed out that the majority of video-arcade game players and designers are young males. There are women players of any game, but popular choices among women for early video games were ‘Pacman’ and its variants, plus a few other games such as ‘Donkey Kong’ or ‘Tetris’. We can only speculate as to why woman prefer these games. One female reviewer labelled Pacman ‘oral aggressive’ and could appreciate the female style of play. Other women have identified the compulsive cleaning up of every dot as an attraction. These games are distinguished by their less violent action and sound track. Also, the board is fully visible, characters have personality, softer colour patterns are used, and there is a sense of closure and completion. Can these informal conjectures be converted to measurable criteria and then validated? Can designers become more aware of the needs and desires of women, and create video games that will be more attractive to women than to men?

Turning from games to office automation, the largely male designers may not realize the effect on women users when the command names require the users to KILL a file or ABORT a program. These and other potentially unfortunate mistakes and mismatches between the user interface and the user might be avoided by paying more thoughtful attention to individual differences among users.

2.6 Age

Historically, computers and computer applications have been designed for use by adults for assisting them in their work. In many accepted definitions of human-computer interaction and interaction design, there is a hidden assumption that the users are adults. In definitions of HCI there are, for example, references to users' 'everyday working lives' or the 'organisation' they belong to. Nowadays, however, computer users span all ages. Applications are developed for toddlers aged two or three and special applications and mobile devices are designed for the elderly. User groups of different ages can have vastly different preferences with regard to interaction with computers.

The average age of the user population affects interface design. It is an indication of the amount of expertise that may be assumed. In many instances, it affects the flexibility and tolerance of the user group. This does not always mean that younger users will be more flexible. They are likely to have used a wider range of systems and may have higher expectations. Age also determines the level of perceptual and cognitive resources to be expected from potential users. By this we mean that our ability to sense (perception) and process (cognition) information firstly develops and then declines over time. Many user interfaces fail to take these factors into account.

Below we look at two special user groups – young children and the elderly – in detail.

2.6.1 Young Children

Child-computer interaction has emerged in recent years as a special research field in human-computer interaction. Children make up a substantial part of the larger user population. Whereas products for adult users usually aim to improve productivity and enhance performance, the purpose of children's products is more likely to provide entertainment or engaging educational experiences. Applications designed for use by children in learning environments have completely different goals and contexts of use than applications for adults in a work environment (Inkpen, 1997). While adults' main reasons for using technology are to improve productivity and to communicate, children do it for enjoyment. Another reason for distinguishing between adult and child products is young children's slower information processing skills that affect their motor skills and consequently their use of the mouse and other input devices (Hutchinson et al., 2007).

Computer technology makes it possible for children to easily apply concepts in a variety of contexts (Roschelle et al., 2000). It exposes them to activities and knowledge that would not be possible without computers. For example, a young child who cannot yet play a musical instrument can use software to compose music. People opposed to the use of computers by young children have warned against some potential dangers. These include keeping children from other essential activities, causing social isolation and reduced social skills, and reducing creativity. There is general agreement that young children should not spend long hours at a computer, but computers do stimulate interaction rather than stifle it. Current advances in technology make it possible to create applications that offer highly stimulating environments and opportunities for physical interaction. New tangible and robotic interfaces are changing the way children play with computers (Plowman and Stephen, 2003). The term 'computer' in child-computer interaction refers not only to the ordinary desktop or notebook computer, but also to programmable toys, cellular phones, remote controls, programmable musical keyboards, robots, and more (see Figure 2.3).

Figure 2.3 Children interacting with the robot QRIO (Swaminathan, 2007)

One way to address the concerns about the physical harm in spending too much time passively in front of a computer screen is to develop technology that require children to move around. Dance mats that use sensory devices to detect movement are widely available. Computer vision and hearing technology can also be used to create games that use movement as input. A widely used commercial application that uses movement input is Sony's EyeToy™. The EyeToy is a motion recognition USB camera used with Sony's Playstation 2. It can detect movement of any part of the body, but most EyeToy games involve arm movements. An image of the player is projected on the screen to form part of the gamespace (see Figure 2.4). Depending on the game context, certain areas of the screen are active during the game. Players must move so that their hands on the projected image interact with screen objects that are active in the game. For example, they have to hit or catch a moving ball. In other words, the user manipulates screen elements through his or her projected image.

Figure 2.4 Projected images of children playing Sony EyeToy games (Game Vortex, 2008)

Clearly, technology has become an important element of the context in which today's children grow up and it is important to understand its impact on children and their development. According to Druin (1996) we should use this understanding to improve technology so that it supports children optimally. The development of any technology can only be successfully if the designers truly understand the target user group. Knowledge of children's physical developmental and familiarity with the theories of children's cognitive development is thus essential when designing for them. The way children learn and play, the movies and television programmes they watch, and the way they make friends and communicate with others, are influenced by the presence of computer technology in their everyday lives. For this reason Druin (1996) believes it is critical that designers of future technology observe and involve children in their work. When designing for children, the important thing is to accommodate children so that they can perform activities on the computer that are at their level of development

2.6.2 The Elderly

Due to advances in health care technologies and living standards the human life span is constantly increasing. This means that the population of older people is growing and that older people are more active than before. Although people are now living longer, many of them will stilll develop some degenerative disabilities due to advanced age (Darzentas and Miesenberger, 2005).

The elderly have often been ignored as users of computers since they are assumed to be both dismissive of and unable to keep up with advancing technology. The stereotype that senior citizens are averse to the use of new technologies is not necessarily true (Dix et al., 2004). They do however experience impairments related to deterioration of vision, movement and memory capacity (Kaemba, 2008) that affect the way they interact with devices. They have problems with mouse use because they complete movements slowly and have difficulty in performing fine motor actions such as cursor positioning. Moving the mouse cursor over small targets may be difficult for senior users, and double clicking actions may be problematic, especially for users with hand tremor.

Dexterity of the fingers decreases as we age, so elderly users may experience many difficulties typing long sequences of text on a keyboard. Keyboards that can easily be reached, have sufficient space between keys, provide audible or tactile feedback of pressed keys, and provide high contrast between text and background may be required. Most of the mechanisms for supporting users with motor impairments described in section 2.3.2.2 are applicable to elderly users.

Many senior users find the text size on typical monitors too small, and require more contrast between text and background. Even more so on small displays of mobile phones. Touch screens solve some of the interaction problems, but older users’ habit of a finger along a text line while reading, can result in unintended selections (Kaemba, 2008). Clearly, the physical, social and mental contexts of the elderly differ from that of younger adults. The needs and preferences of adult technology users can therefore not always be generalised to the elderly.

ACTIVITY 2.6

Identify two cell phone users aged 15 or younger and two aged 65 or older. Ask each of them to list three things they like about their cell phones and three things they do not like.

By comparing the lists, can you identify differences in the needs and preferences of users from the different age groups?

2.7 Expertise

The way in which a system is designed, built, and sold differs if the intended users are 'experts' or 'novices'. In the former case, designers must build upon existing skills. Issues such as consistency with previous interfaces are absolutely critical. In the case of novices, designers must provide a higher level of support. They must also anticipate some of the learning errors that can arise during interaction. It is difficult to begin the development process if designers are unaware of such general characteristics of their user population. Some people may only have partial information about how to complete a task. This, typically, is the situation of a novice user of a computer application. They will need procedural information about what to do next. Experts will have well-formed task models and may not need this guidance. It follows, therefore, that for novel tasks designers may have greater flexibility in the way that they implement their interface. In more established applications, expert users will have well-developed task structures and may not notice or adapt so quickly to any changes introduced in a system.

A number of models of skill level have been developed to provide an explanation of how users operate at the different levels. The one in Figure 2.5 shows the differences between users with different degrees of information about an interactive system. At the lowest level, the knowledge-based level, they may only be able to use general knowledge to help them understand the system. Designers can exploit this to support novice users. For example, in the Windows desktop, inexperienced users can apply their general knowledge in several ways, but sometimes with an unwanted effect. To recover a deleted file a user might think he has to 'empty the recycle bin (waste bin)'. This is a dangerous approach, however. If knowledge fails then users are forced to make guesses.

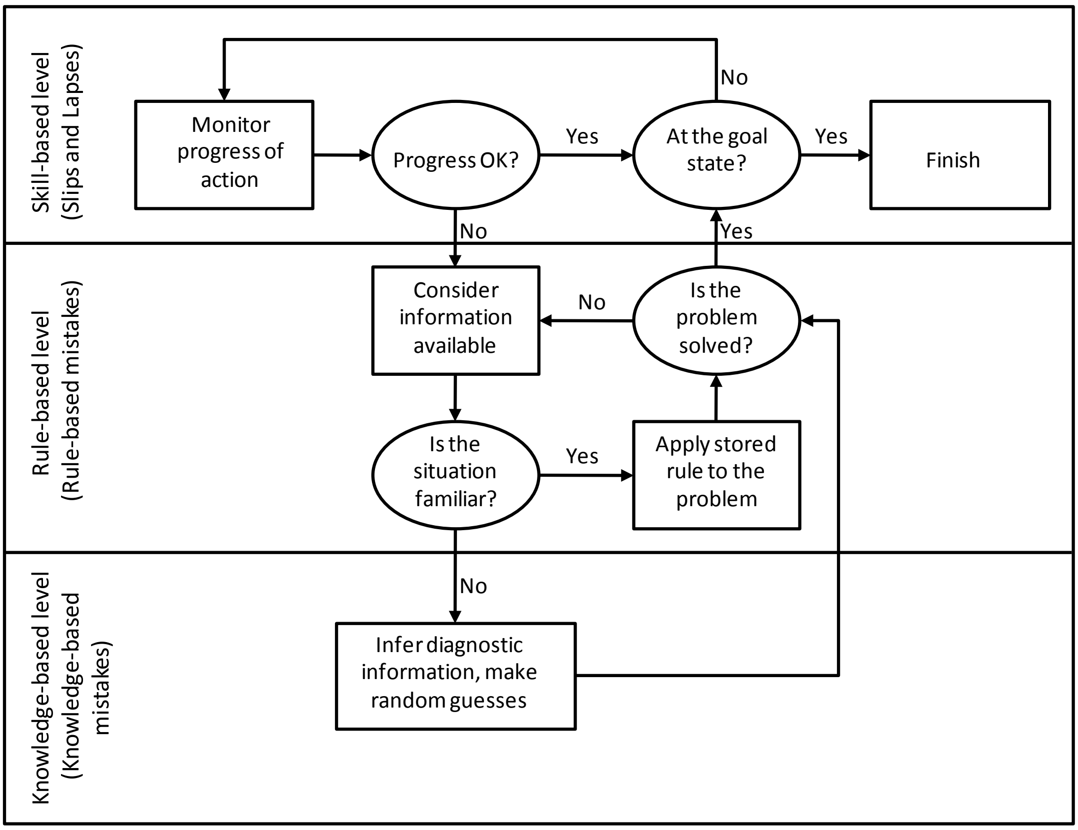


Figure 2.5 Three levels of expertise

The second level of interaction introduces the idea that users exploit rules to guide their use of a system. This approach is slightly more informed than the use of general knowledge. For example, users will make inferences based on previous experience. This implies that designers should develop systems that are consistent. Similar operations should be performed in a similar manner. If this approach is adopted then users can apply the rules learned with one system to help them operate another. 'To print this page, I go to the File menu and select the option labelled Print'. There are two forms of consistency:

- *Internal consistency* refers to similar operations being performed in a similar manner within an application. This is easy to achieve if designers have control over the finished product.
- *External consistency* refers to similar operations being performed in a similar manner between several applications. This is hard to achieve as it involves the design of systems in which the designer may have no involvement. This is the reason why companies such as Apple and IBM publish user interface guidelines.

Operating a user interface by referring to rules learned in other systems can be hard work. Users have to 'work out' when they can apply their expertise. It also demands a high level of experience with computer applications. Over time users will acquire the expertise that is required to operate a system. They will no longer need to think about previous experience with other systems and will become skilled in the use of the system. This typifies expert use of an application (the skill-based level in Figure 2.5).

What designers should always keep in mind is that the more users have to think about using the interface the less cognitive and perceptual resources they will have available for the main task.

2.8 The Errors People Make

2.8.1 Types of Errors

People make errors routinely. It is part of human nature. There are several forms of human error or mistakes. Norman (1999) distinguishes the following main categories:

- Mistakes (also called incorrect plans): This category includes incorrect plans such as forming the wrong goal, or performing the wrong action with relation to a specific goal. Situations in which operators adopt unsafe working practices are examples of this. These can arise either through lack of training, poor management or deliberate negligence. Mistakes are thus the result of conscious, but erroneous, consideration of options.
- Slips: Slips are observable errors and result from automatic behaviour. They include confusions such as between left and right.

So with a slip the person had the correct goal but performed the incorrect action; with mistake the goal was incorrect.

Slips occur mostly in skilled behaviour, when the person is not paying proper attention. While still learning to do something, a people don't make slips (Norman, 1999).

Norman (1999) distinguishes between the following kinds of slips:

- Capture errors: This occurs when an activity that you perform frequently is done instead of the intended activity. For example, when I take a day's leave and still drop my child at the pre-school, I drive from the school to work instead of home.

- Description errors: This occurs when instead of the intended activity, you do something that has a lot in common with what you wanted to do. For example, instead of putting the ice-cream in the freezer you put it in the fridge.
- Data-driven errors: These errors are triggered by some kind of sensory input. I asked the babysitter to write her telephone number in my telephone directory. Instead of her own number she copied the number of the entry just above where she wrote hers. She was looking at that entry to see whether that person's name or surname was written first.
- Mode errors: These occur when a device has different modes of operation and the same action has a different purpose in the different modes. For example, a watch can have a time-reading mode and a stopwatch mode. If the button that switches on a light in time-reading mode is also the button that resets the stopwatch, one may try to read the stopwatch in the dark by pressing the light button and thereby accidentally clear the stopwatch.
- Associative activation errors: These are similar to description errors but are triggered by internal thoughts or associations instead of external data. For example, our secretary's name is Lynette, but she looks a lot like someone I know called Irene. I often call her by the name Irene.
- Loss-of-activation errors: These are errors due to forgetting. For example, you find yourself sitting with the phone in your hand but you have forgotten who you wanted to call.

ACTIVITY 2.7

1. Give examples from your own experience or environment of each of the following: capture error, description error, data-driven error, mode error, associative activation error, and loss of activation error.

2. Do research to find information on each of the following disasters. Can you identify the errors made as any of the types discussed above?
   - The Three Mile Island disaster
   - The Chernobyl disaster
   - Kegworth aircraft crash
   - Mars climate orbiter disaster.

2.8.2 The Cause of Human Error

What is the true cause of human error? In the aftermath of many major accidents, it is typical to hear reports of 'operator error' as the primary cause of the failure. This term has little meaning unless it is supported by a careful analysis of the accident. For example, if the operator is forced to manage as best he/she can with a bug-ridden, unreliable system, then is an accident his/her fault or that of the person who implemented the program? In turn, if the bugs are the product of poorly defined requirements or cost cutting during testing, then are these failures the fault of the programmer or the designer?

Often, what appears to be operator error is the result of management failures. Even if systems are well designed and implemented, accidents can be caused because operators are poorly trained to use them. This raises practical problems in that operators are frequently ill-equipped to respond to the low-frequency, high-cost errors. How then can companies predict these events that occur rarely, yet which are sufficiently critical that users and operators should be trained in the procedures to use them?

Further sources of error come from poor working environments. Again, a system may work well in a development environment but the noise, heat, vibration, or the altitude of a user's daily life, may make the system unfit for its actual purpose.

2.8.3 How to prevent human error

There is no simple way to improve the operational safety of computer systems. Short term improvements in operator training will not address the fundamental problems created by mistakes and slips. Reason (1990) argues that errors are latent within each one of us and, therefore, we should never hope to engineer out human error. This pessimistic analysis has been confirmed by experience. Even organisations such as NASA, with an exemplary training and recruitment system, have suffered from the effects of human error.

There are, however, some obvious steps that can be taken to reduce both the frequency and the cost of human error. In terms of cost, it is possible to engineer decision support systems that provide users with guidance and help during the performance of critical operations. These systems may even implement 'cool off' periods during which users' commands will not be effective until they have reviewed the criteria for a decision. These systems engineering solutions actively impose 'interlocks' to control and limit the scope of human intervention. The consequences are obvious when such locks are placed upon inappropriate areas of a system.

It is also possible to improve working practices. Most organisations see this as part of an on-going training program. In safety-critical applications there may be continuous and on-the-job competence monitoring, as well as formal examinations.

When designing systems one should keep the kind of errors people make in mind. For example, minimising different modes or making the different modes clearly visible, will avoid mode errors. Users may click on a delete button when they meant to click on the save button (maybe the delete button is located where, in a different application, the save button was placed). To prevent the user from incorrectly deleting something important, the interface should request confirmation before going through with a delete action.

ACTIVITY 2.8
Using only information from this unit of the study guide, identify fifteen guidelines for the design of usable and/or accessible interfaces. Formulate them in your own words and in a way that will be useful to designers.

2.9 Conclusion

Most computer-based systems are developed for use by people. Different systems are aimed at different kinds of people. It is crucial that whoever design these systems understand who they are designing for and how interacting with the system may affect these users.

In this unit we tried to give you a taste of the complexity of human cognition and human nature. Our goal was to help you realise that human problems and errors with technology are often a design failure, and that good design takes human capabilities and weaknesses in account.

The final conclusion: there is no such thing as the average user.

NOTES

UNIT 3
Design Problems and Solutions

CONTENTS

UNIT 3 - OUTCOMES

After studying this unit you should:

- Understand the problems of designing and interactive system, and specifically:
  - How natural evolution of design is obstructed.
  - Common mistakes designers make
- Understand and be able to apply the design practices and tools to avoid the problems with the design of interactive systems, namely:
  - Mechanisms such as affordance, constraints, mapping, visibility and feedback.
  - Design guidelines, principles and standards.

3.1 Introduction

Now that we have identified the key characteristics that distinguish different users, we will discuss some of the more concrete ways to make design decisions. We start by looking at some of the most common problems of interface design and then discuss how design problems can be overcome. We look at guidelines and principles for design compiled by some of the most influential researchers and authors in the field of HCI.

A substantial part of this unit is based on the work of Don Norman, as presented in his most important book – The Design of Everyday Things. The book was first published in 1989 and the 2000 edition hasn't changed much. Although 1989 computer technology looked vastly different from what we know today, all the principles and advice given by Norman still apply. Reading the book will help you in this module and will change the way you look at everyday objects.

3.2 Design Problems

There are, potentially, many mistakes that designers can make, but Norman (1999) points out the following as being the most problematic: the forces that work against evolutionary design, putting aesthetics first (that is, form over function) and designers regarding themselves as typical users.

3.2.1 Hampering the Natural Evolution of Design

Norman (1999) coined the term 'evolutionary design' that refers to the process whereby a product is gradually improved over time. Evolutionary design occurs when a design evolves through a cycle of testing, identifying problems, modification, re-design, re-testing, re-modification, and so on, until a functional, aesthetically pleasing object results. Good features are kept unchanged while bad features are removed or replaced with improved versions. This is a good process, but, unfortunately there are obstacles to such natural evolution. Three forces that work against evolutionary design are (Norman, 1999):

1. The demands of time

New versions of an object are already in the process of being designed before the old one has been released. Hence, even if someone took the trouble to get feedback from users of the old version, there would not be time to adapt the new design to address possible problems with the previous one. The fact that Microsoft often releases a new version of their operating system when there are still problems with it, is an example where releasing it on the promised date is more important than providing customers with a bug-free application. Hence the need for 'service packs' and 'hot fixes'.

2. Pressure to be distinctive

Each design must have features that distinguish it from previous models so that consumers can be lured with statements like 'a new improved version'. Often the new model doesn't even incorporate the good qualities of its predecessor.

3. The curse of individuality and market differentiation

Every company that manufactures the same type of product has to come up with a unique design which carries their signature. This means that if one company perfects a product, all the others that produce it will create an inferior product in the name of individuality. Of course, the quest for individuality can also lead to innovative solutions to real problems, but the goal should be to improve the product or solve the problem, not just to stand out.

Have you ever wondered why the keys on a computer keyboard are arranged in the order QWERTY....? On the first rectangular typewriter keyboard, the keys were arranged alphabetically. This allowed for typing speeds that were too much for the typewriter's mechanics – if the typing was too fast the parts would get jammed. The solution was to rearrange the keys in a way that would slow down the typist. Here a natural evolutionary process was followed, but the main driving force was the mechanical limitations of the instrument. People got so used to using the QWERTY keyboard, that it is still used even if it was designed under constraints that have disappeared long ago.

ACTIVITY 3.1
Discuss any example of a design that reflects evolutionary design in the good sense, or that demonstrates how the forces against evolutionary design have prevented a product from naturally evolving into something better.

3.2.2 Common Design Mistakes

In unit 2 we addressed the broad mistake of designing for the typical user. Here we consider two more issues, namely overestimating the importance of aesthetics and designers regarding themselves as typical users. We then briefly look at the problem of cluttered interfaces.

3.2.2.1 Putting Aesthetics above Usability

We cannot argue against the fact that part of the appeal of Apple products is how they look. From the start Apple Macintosh paid special attention to the aesthetics of their products. Aesthetics should, however, not be the determining factor to the detriment of usability. Not long ago, computer applications could only be produced by computer scientists. Nowadays the development tools allow people with limited or no programming knowledge to create applications such as web pages. The competitive commercial environment provides good motivation to employ graphic designers and artists to create attractive interfaces. Unfortunately, these designers do not always understand the importance of usefulness and usability.

An interface need not be an artwork to be aesthetically pleasing. One that is free of clutter, with the interface elements organised in a logical and well-balanced way, and that uses colour tastefully can provide a lot of visual pleasure to users who have to find their way through the interface.

Here again the target user group should be considered. Young children prefer colourful interfaces with icons that move or twirl when the mouse moves over them, but this will annoy most older users. Culture may also determine what the user finds aesthetically pleasing.

Google.com is proof that a beautiful interface is not a prerequisite for a successful system. Google’s interface is pretty simple, but no one should find it offensive or unusable.

3.2.2.2 Thinking for the User

Designers sometimes believe that they know what the user would want, thinking that they can put themselves in the shoes of the user. Designers are expert users of technology and they most often design applications that will be used by people who have far less knowledge of, and exposure to, technology. People tend to project their own feelings and beliefs onto others (e.g. mothers who force their children to wear jerseys because they themselves are cold). Designers are no different – they will subconsciously build interfaces according to their own preferences and knowledge.

When a system is complete, the designers and developers know it so well that they will never be able to view it from the perspective of someone who encounters it for the first time. The users’ model of a system will be very different from that of a system designer. Users’ view of an application is heavily influenced by their tasks and by their goals and intentions. For instance, users may be concerned with letters, documents and printers. They are typically less concerned about the disk scheduling algorithms and device drivers that support their system.

Clearly, if designers continue to think in terms of engineering abstractions rather than the objects and operations in the users’ task, then they are unlikely to produce successful interfaces. It is essential for designers to realise that they will make this mistake if they do not involve real users in design process. The earlier in the process this happens, the better.

Another common error is to mistake the client for the end user and base the designs on the requirements specified by the client. For example, a university’s management may decide that they need a web-based learning management system that students can use to find information about their courses, download study material and communicate with their lecturers and fellow students. They employ an IT development company to design, develop and implement the system according to their (management’s) specifications. The designers should first determine who the end users will be (in this case the students) and test the specifications provided by university management against the requirements and preferences of these users.

3.2.2.3 Cluttering the Interface

Interfaces should provide users with enough information to allow them to perform their required task successfully. They should, however, avoid screen clutter. One reason is that it affects the aesthetics of the design, but more important reasons are:

- It can be difficult for users to take in and understand the many different objects that are presented on the screen. Some may be missed entirely.

- The more objects you present on the screen at once, the more meanings users will have to unravel.
- The more objects you present, the harder it is for users to find the ones that they really need.
- The more objects there are on the screen, the smaller the average size of each object will be. This makes it harder to select and manipulate individual screen components.

3.3 Design Solutions

In section 3.2 we highlighted some common design problems and gave some advice on how to avoid them. On a more positive note, this section will offer a range of mechanisms that support successful interface design. Principles for good interface design are not unique to interactive computer systems. Most basic principles for good interface design can be derived from Norman's (1999) good principles for the design of everyday things. Below we describe his well-known and widely accepted design concepts.

3.3.1 Affordance

Affordance is a property of an object or interface that refers to its perceived and actual properties that tells an observer or user how the object or interface can be used. For example, the handles of a pair of scissors usually have one hole that is smaller than the other – one round and one oval shaped. This tells the user how to hold the scissors. The thumb goes through the smaller hole while the other fingers go through the bigger hole. The stronger the clues provided by the affordances the better the user will know what to do.

Consider the interface below:

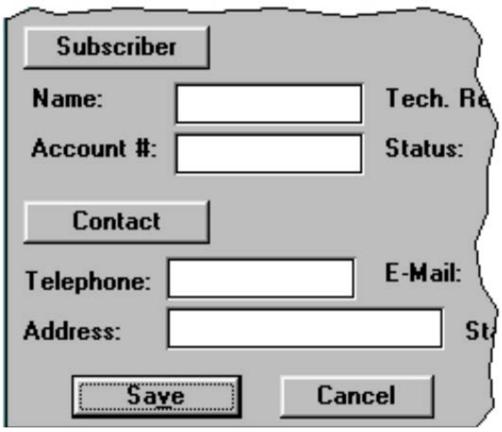


Figure 3.1 Example of affordance

The Subscriber and Contact buttons in this interface fulfil the purpose of headings. Clicking on them will have no effect. Still, the user will think they should click on these 'buttons' since they invite clicking. So, we should use affordance to guide the user into taking the correct action, but we should also be careful to use controls for some purpose if they clearly afford another.

The Save and Cancel buttons on this interface also invite clicking, as they should.

3.3.2 Constraints

A constraint in HCI terms is a mechanism that restricts the allowed behaviour of a user when interacting with a computer system. For example, an ATM will only accept your card if you insert it into the slot the right way round. This is a physical constraint – it relies on properties of the physical world for its use. If a user cannot interpret the constraint easily, they will still have difficulty to perform the action. Often ATMs have a small icon next to the insertion slot to indicate to the user how the card should be inserted.

Not all constraints are physical. Constraints can also rely on the meaning of the situation (semantic constraint) or on accepted cultural conventions (cultural constraints). The fact that a red traffic light constrains a driver from crossing the road is an example of a semantic constraint – the driver knows that a red light means he should stop if he wants to prevent an accident. He is not physically forced to stop, but his interpretation of the situation makes him stop.

In some cultures it is customary for a man to stand back to let a woman enter through a door first. Men who follow this custom are 'constrained' by the cultural convention. An example of using a cultural constraint in interface design is to use a green button to go ahead with an operation or action and a red button to indicate the opposite. This follows the cultural convention that red means 'stop' or 'danger' while green means 'go' or 'ok'.

Logical constraints refer to constraints that rely on the logical relationships between functional and spatial aspects of the situation. Suppose there are two unmarked buttons on the doorbell panel at a house you are visiting. If you have no knowledge of the house or the people who live there it will be difficult to decide which the correct button is to push. If you know that there is a flat that is situated to the left of the house you can assume that the left hand button is that for the flat and the right hand one for the house (assuming the occupants used some logic when installing the system). Natural mappings work according to logical constraints.

A forcing function is a type of physical constraint that requires one action before a next can take place (Norman, 1999). The ATM example above is an example of a forcing function. Another example is that you cannot switch on a front loading washing machine unless the door is properly closed.

3.3.3 Mapping

Mapping refers to relationship between two things, for example, the relationship between a device's controls and their movements, and the results of the actual use of these controls. A good mapping is a mapping that enables users to determine the relationships between possible actions and their respective results. Programming your television's channels so that you get SABC1 by pressing 1 on the remote control, SABC2 by pressing 2, SABC3 by pressing 3 and e-TV by pressing 4 is an example of a natural mapping. In a computer interface there should be good mapping between the text on buttons or menus and the functions activated by choosing those buttons or menu items. A standard convention in Windows interfaces is to use 'Save' on a menu item that overwrites the current copy of a document and Save As ... when you want to create a new instance of the document. By now most people are familiar with this, but it would have been a better mapping to name the menu item 'Save a Copy'.

Natural mappings use physical analogy and cultural standards to support interpretation. Figure 3.2 shows some icons from a children's game. The page backward and page forward icons provide a natural mapping with their functions. They clearly depict a page and the arrows indicate the direction of paging through the document. Their spatial orientation further strengthens the mapping – the left hand one for backwards and the right hand one for forwards. The traffic light icon, which is for exiting the page, does not. There is no logical, spatial or semantic connection between a traffic light and the exit operation.

Figure 3.2 Icons from a children's game

3.3.4 Visibility

The parts of a system that is essential for its use must be visible. The visible structure of well-designed objects gives the user clues about how to operate them. These clues take the form of affordances, constraints and mappings. Visible signs (like letters or the colour) on salt and pepper shakers tell us which is which. The main menu of Storybook Weaver Deluxe 2004 is given in Figure 3.3 (the explanatory text provided in this figure does not appear on the interface). The absence of text labels to the icons makes it difficult for users to interpret them – especially young children who will not associate a light bulb with 'story ideas' or a feather with 'creating a new story'. There are in fact text labels associated with the icons, but they only become visible if the mouse pointer is moved across the icons. There is, however, no way for users to know this. This interface fails badly in terms of visibility.

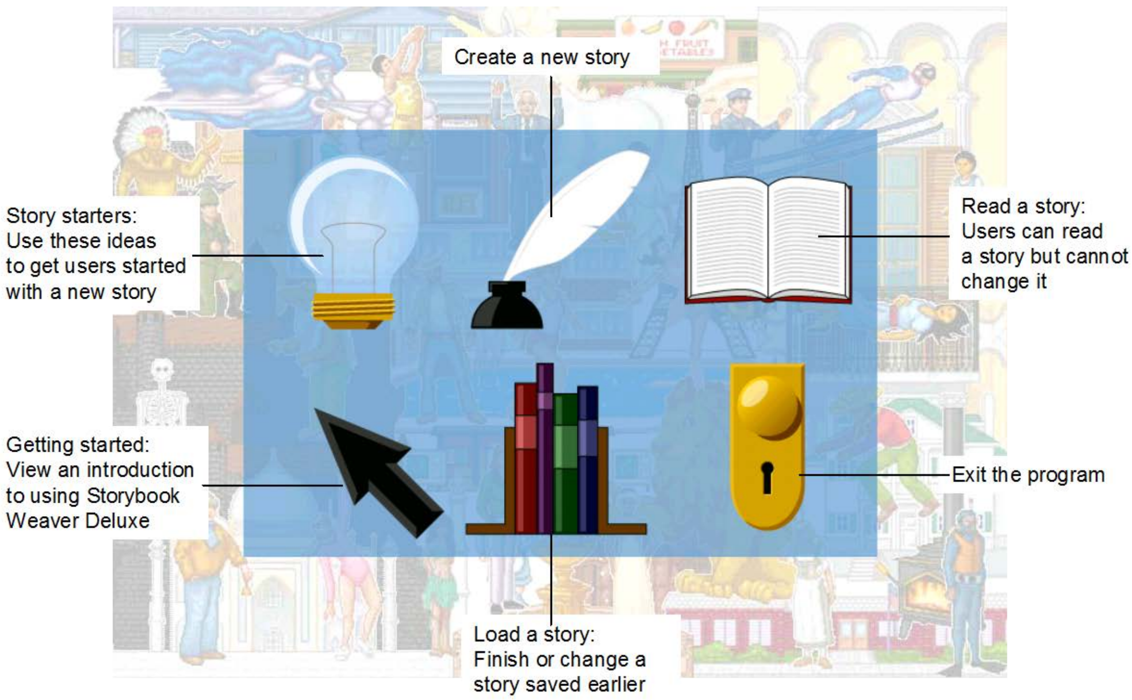


Figure 3.3 Opening screen of Storybook Weaver Deluxe 2004

Sound can also used to make interface elements more 'visible'. Often an error message has a sound attached to it to draw the user's attention to the problem. In products for children who cannot yet read, audio cues can be attached to icons instead of text labels. Sound calls our attention to an interface when there is new information, e.g. a beep on a cell phone signalling the arrival of a new message.

3.3.5 Feedback

Feedback is information that is sent back to the user about what action has actually been performed, and what the result of that action is. When typing we know that we have pressed the keys hard enough if the letters appear on screen. Operations that take time are often indicated by a progress bar or a message stating that the process is under way. Without constant feedback the interaction process will be very unsatisfactory.

Novices want more informative feedback to confirm their actions; frequent users want less distracting feedback.

Consider Figure 3.4 that shows a window that appears directly after a user of Storybook Weaver Deluxe 2004 has clicked on the 'Save As Web Document' option on the File menu. The web document is automatically saved in the user's My Documents folder in a subfolder called Storybook Weaver Deluxe. The title of the story is used as the name of the web document. Is this suitable and adequate feedback for a young child?

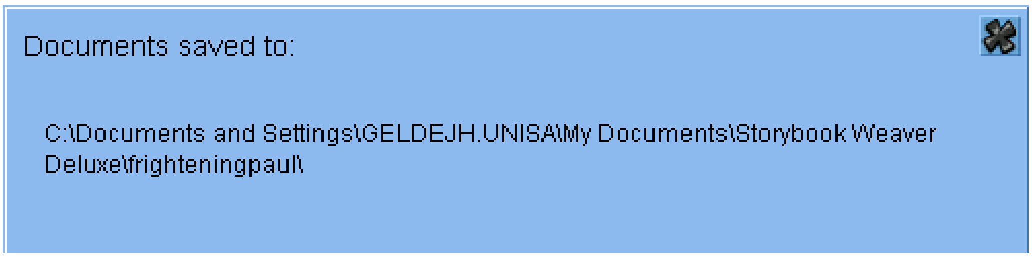


Figure 3.4 Feedback after saving a story as a web document

ACTIVITY 3.2
Suggest ways in which the feedback in figure 3.4 can be made more suitable for a six or seven-year-old user.

Sound is an important feedback mechanism. We know the washing machine's door is closed properly when we hear the click sound. Sound feedback is extensively used in cars – my car beeps annoyingly at a steadily increasing volume when I drive without fastening my seatbelt, it beeps when the petrol tank is close to empty, and so on. The absence of sound can also be a form of feedback. If you switch the kettle on and do not hear the water heating up after a while it is an indication that something is wrong.

ACTIVITY 3.3

After a tiring journey, you check into a guest house. In your room there is a welcome tea tray and kettle. But, oh dear - the kettle cord dangles and there's no sign of an electric wall plug near the kettle. After filling the kettle in the bathroom, you go down on hands and knees and find an extension cord coming from behind a cupboard. Relief.... you plug in the kettle, but the kettle has an up-down switch and the extension has an embedded left-right switch. Neither switch indicates ON or OFF and there are no friendly little red lights to show that current is flowing. By trial and error you work out the right combination. The tea, eventually, was really good.

Referring to Norman's principles of design (referring to affordance, mapping, constraints, visibility and feedback), clearly identify which of these concepts are relevant to the given case, define them and then show how they were implemented or not implemented in this situation.

3.3.6 Guidelines, Principles and Standards

The aim with design guidelines, standards and design principles is to help designers to improve the usability of their products by giving them rules according to which they can make design decisions (Dix et al., 2004). Such rules restrict the range of design options and prevent the designer from making choices which are likely to harm the usability of the product. Dix et al. classify design rules as standards or guidelines. Standards are usually set by national or international bodies, are high in authority and limited in application, while guidelines are more general in application.

There are two types of design guidelines: low-level detailed rules and high-level directing principles. High-level principles are relatively abstract and applicable to different systems, while detailed rules are instructions that are application-specific and do not need much interpretation.

The difference between design principles and usability principles are that design principles usually informs the design of a system, while usability principles are mostly used as the basis for evaluating prototypes and complete systems (Preece et al., 2007). Usability principles can be more prescriptive than design principles. When used in practice, some design or usability principles are referred to as heuristics (Preece et al., 2007).

Below we discuss some of the most prominent sets of guidelines, namely those of Dix et al. (2004), Preece et al. (2007), and Shneiderman (1998).

3.3.6.1 Dix, Finlay, Abowd and Beale

Dix et al. (2004) provide interface designers with a comprehensive set of high-level directing principles with the aim of improving the usability of interactive systems. They divide their principles into three categories, namely Learnability principles, Flexibility principles and Robustness principles. They summarise their principles in three tables that we reproduce below as tables 3.1, 3.3 and 3.5. In these tables our added explanations appear in italics. The related principles are described in tables 3.2, 3.4 and 3.6 respectively.

Learnability

Learnability refers to the ease with which users can enter a new system and reach a maximal level of performance (Dix et al., 2004). Dix et al. identified five principles that affect the learnability of a computer-based system. They are defined in Table 3.1.

Table 3.1 Principles that affect Learnability (from Dix et al. (2004), p. 261)

Principle	**Definition**	**Related principles (explained in Table 3.2)**
Predictability	Support for the user to determine the effect of future action based on past interaction history.	Operation visibility
Synthesisability	Support for the user to assess the effect of past operations on the current state. *To be able to predict future behaviour, a user should know the effect of previous actions on the system. Changes to the internal state of the system must be visible to users so that they can associate it with the operation that caused it.*	Immediate/eventual Honesty
Familiarity	The extent to which a user's knowledge and experience in other real-world or computer-based domains can be applied when interacting with a new system. *The user's first impression is important here. Familiarity can be achieved through metaphors and through effective use of affordances that exist for interface objects. Clickable objects must look clickable, for example.*	Guessability, affordance
Generalisability	Support for the user to extend knowledge of specific interaction within and across applications to other similar situations.	
Consistency	Likeness in input-output behaviour arising from similar situations or similar task objectives.	

We expand on the related principles mentioned above in Table 3.2.

Table 3.2 Principles that relate to Learnability principles

Principle	**Explanation**
Operation visibility	The way in which the availability of possible next operations are shown to the user and how the user is informed that certain operations are not available.
Honesty	The ability of the user interface to provide an observable and informative account of any change an operation makes to the internal state of the system. It is immediate when the notification requires no further interaction by the user. It is eventual when the user has to issue explicit directives to make the changes observable.
Guessability and affordance	The way the appearance of the object stimulates a familiarity with its behaviour or function.

Flexibility

Flexibility refers to the many ways in which interaction between the user and the system can take place. Dix et al.'s (2004) main principles of flexibility are explained in table 3.3 and other principles that relate to these, in table 3.4.

Table 3.3 Principles that affect Flexibility (from Dix et al. (2004), p. 266)

Principle	Definition	Related principles
Dialogue initiative	Allowing the user freedom from artificial constraints on the input dialogue imposed by the system.	System/user pre-emptiveness
Multi-threading	Ability of the system to support user interaction pertaining to more than one task at a time.	Concurrent vs. interleaving, modality
Task migratability	The ability to pass control for the execution of a given task so that it becomes either internalised by user or system or shared between them. *For example, a spell checker does some of the work but should ultimately let the user decide which words to replace.*	
Substitutivity	Allowing equivalent values of input and output to be arbitrarily substituted for each other.	Representation multiplicity, equal opportunity
Customisability	Modifiability of the user interface by the user or the system.	Adaptivity, adaptability

Table 3.4 Principles that relate to Flexibility principles

Principle	Explanation
System pre-emptiveness	This occurs when the system initiates all dialogue and the user simply responds to requests for information. It hinders flexibility, but may be necessary in multi-user systems where users should not be allowed to perform actions simultaneously.
User pre-emptiveness	This gives the user freedom to initiate any action towards the system. It promotes flexibility, but too much freedom may cause the user to lose track of uncompleted tasks.
Concurrent/ interleaved multi-threading	Concurrent multi-threading allows simultaneous communication of information pertaining to separate tasks. Interleaved multi-threading permits temporal overlap between separate tasks, but at any time the dialogue is restricted to a single task.
Multi-modality	Separate modalities (channels of communication) are combined to form a single input or output expression.
Representation multiplicity	Flexibility for rendering of state information, e.g. in different formats or modes.
Equal opportunity	Blurs the distinction between input and output at the interface – the user has the choice of what is input and what is output; in addition, output can be reused as input.
Adaptability	Refers to user-initiated modification to adjust the form of input and output. Users may for example choose between different languages or complexity levels.

Adaptivity	Refers to system-initiated modification to customise the user interface automatically. Here the system should observe the users' behaviour (for example, repeated attempts at tasks) and determine their level of expertise in order to adjust the complexity level of tasks.

Robustness

Robustness refers to the level of support the user is given for successful achievement and assessment of their goals (Dix et al., 2004). Table 3.5 summarises Dix et al.'s Robustness principles and table 3.6 lists the supporting principles.

Table 3.5 Principles that affect Robustness (from Dix et al. (2004), p. 270)

Principle	Definition	Related principles
Observability	Ability of the user to evaluate the internal state of the system from its perceivable representation. The user compares the current state with his or her intention within the task-action plan.	Browsability, static/dynamic defaults, reachability, persistence, operation visibility
Recoverability	Ability of the user to take corrective action once an error has been recognized.	Reachability, forward/backward recovery, commensurate effort
Responsiveness	How the user perceives the rate of communication with the system. Response time is the duration of time needed by a system to inform the user of state changes. When this is not instantaneous the system should give some indication that the task is in progress.	Stability
Task conformance	The degree to which the system services support all of the tasks the user wishes to perform and in the way the user understands them.	Task completeness, task adequacy.

Table 3.6 Principles that relate to Robustness principles

Principle	Explanation
Browsability	This allows the user to explore the current internal state of the system via the limited view provided at the interface. The user should be able to browse to some extent to get a clear picture of what is going on, but negative side-effects should be avoided.
Static/Dynamic defaults	Static defaults are defined within the system or acquired at initialisation. Dynamic defaults evolve during the interactive session (for example, the system may pick up a certain user input preference and provide this as the default input where applicable).
Reachability	The possibility of navigation through the observable system states.
Persistence	Deals with the duration of the effect of a communication act and the ability of the user to make use of that effect. Audio communication persists only in the user's memory while visual communication remains available as long as the user can see the display.

Backward recovery	Involves an attempt to undo the effects of previous interaction in order to return to a prior state.
Forward recovery	Involves the acceptance of the current state and negotiation from that state towards the desired state.
Commensurate effort	If it is difficult to undo a given effect on the state, then it should have been difficult to do in the first place.
Stability	The invariance in response times for identical or similar computational resources.
Task completeness	Refers to the coverage of all the tasks of interest and whether or not they are supported in a way the user prefers.
Task adequacy	This addresses the user's understanding of the tasks.

3.3.6.2 Preece, Rogers and Sharp

Preece, et al. (2007) discuss two types of design goals in interaction design, namely usability goals and user experience goals. The usability goals focus on aspects such as effectiveness and learnability, while the user experience goals are concerned with the quality of the user's experience with the system and focus on aspects such as aesthetics and enjoyment.

Usability Goals

Preece et al. (2007) have identified six usability goals that will ensure that users' interaction with technology is effective and enjoyable. We summarise these goals in Table 3.7.

Table 3.7 Preece et al.'s (2007) usability goals

Usability goal	Explanation
Effectiveness	A general goal that refers to how well a system is doing what is what designed for.
Efficiency	This has to do with how well a system supports users in carrying out their work. The focus is on productivity.
Safety	Protecting the user from dangerous conditions and undesirable situations.
Utility	The extent to which a system provides the required functionality for the tasks it was intended to support. Users should be able to carry out all the tasks in the way they want to do them.
Learnability	How easily users learn to use the system.
Memorability	How easy it is to remember how to perform tasks that have been done before.

User Experience Goals

According to Preece et al. (2007), how the user feels about a product irrespective of its efficiency, effectiveness, learnability and so on, plays an important role in it being well accepted or not. For a system to provide users with positive experiences of interaction, designers should attend to features that will make the product satisfying, enjoyable, engaging, pleasurable, exciting, entertaining, helpful, motivating, aesthetically pleasing, supportive of creativity, cognitively stimulating, rewarding, fun, provocative, surprising, emotionally fulfilling, challenging, and enhancing sociability (Preece et al., 2007). Features that make a product boring, frustrating, annoying or overly cute should be avoided.

Clearly one would not spend too much design effort on making a spreadsheet application entertaining or emotionally fulfilling, but these user experience goals are applicable to many new technologies in different application areas. Factors that may support the fulfilment of these user experience goals include attention, pace, interactivity, engagement and style of narrative (Preece et al., 2007).

Design Principles

According to Preece et al. (2007), design principles are prescriptive suggestions to help designers to explain or improve their designs. Instead of telling the designer exactly how to design an interface, they inspire careful design, telling the designer what will work and what not. Preece et al. discuss a number of design principles that we summarise in Table 3.8. You will see that they correspond very well with Norman's principles as discussed in section 3.2.

Table 3.8 Summary of Preece et al.'s (2007) discussion of design principles

Principle	Explanation
Visibility	The more visible the available functions are, the better users will be able to perform their next task.
Feedback	This involves providing information (audio, tactile, verbal or visual) about what action the user has performed and what the effect of that action was.
Constraints	These restrict the actions a user can take at a specific point during the interaction. This is an effective error prevention mechanism.
Mapping	This has to do with the relationships between interface elements and their effect on the system. For example, clicking on a left-pointing arrow at the top left hand corner of the screen takes the user to the previous page and a right-pointing arrow in the right hand corner take the user to the next page.
Consistency	This is similar to consistency as defined by Dix et al. (2004).
Affordance	This refers to an attribute of an object that tells people how it should be used. In an interface it is the perceived affordance of an interface element that helps the user see what it can be used for. Whereas a real button affords pushing, an interface button affords clicking. A real door affords opening and closing, but an image of a door on an interface affords clicking in order to 'open' it.

3.3.6.3 Shneiderman

Shneiderman's (1998) principles for user-centred design are divided into three groups, namely recognition of diversity, golden rules, and prevention of errors.

Recognise Diversity

Before the task of designing a system can begin, information about the intended users, tasks, environment of use and frequency of use must be gathered. According to Shneiderman, this involves the characterisation of three aspects relating to the intended system: usage profiles, task profiles and interactions styles. We explain these in Table 3.9.

Table 3.9 Three aspects relating to recognition of diversity (Shneiderman, 1998)

Aspect	Explanation
Usage profiles	Designers must understand the intended users. Shneiderman lists several characteristics that should be described. Those that apply to young children are age, gender, physical abilities, level of education, cultural or ethnic background, and personality. Designers should find out whether all users will be novices or if they will have experience with the particular kind of system, or if a mixture of novice and expert users are expected. Different levels of expertise will require a layered approach whereby novices are given few options to choose from and are closely protected from making mistakes. As their confidence grows they can move to more advanced levels. Users who enter the system with knowledge of the tasks should be able to progress faster through the levels.
Task profiles	A complete task analysis should be done and all task objects and actions identified.
Interaction styles	Suitable interaction styles should be identified from those available. Here Shneiderman mentions menu selection, form fill-in, command language, natural language and direct manipulation. In Unit 4 we discuss most of the currently available interaction styles.

The Eight Golden Rules for Interface Design

Shneiderman (1998) suggests eight principles of design that are applicable to most interactive systems. They overlap to some extent with those of Dix et al. (2004) and Preece et al. (2007) and are mostly self-explanatory. They are:

1. Strive for consistency.
2. Enable frequent users to use shortcuts.
3. Offer informative feedback.
4. Design dialogues to yield closure (the completion of a group of actions).
5. Offer error prevention and simple error handling.
6. Permit easy reversal of actions.
7. Support internal locus of control (the user should feel in control of the system and not vice versa).
8. Reduce short-term memory load.

Prevent Errors

The last group of principles proposed by Shneiderman (1998) pertain to designing to prevent the user from making errors. Errors are made by even the most experienced users. One way to reduce the loss in productivity due to errors is to improve the error messages provided by the computer system. A more effective approach is to prevent the errors from occurring. The first step towards attaining this goal is to understand the nature of errors (we discussed this in Unit 2). The next step is to organise screens and menus functionally, designing commands and menu choices to be distinctive and making it difficult for users to perform irreversible actions. Shneiderman suggests three techniques which can reduce errors by ensuring complete and correct actions:

- Correct matching pairs: For example, when a user types a left parenthesis, the system displays a message somewhere on the screen that the right parenthesis is outstanding. The message disappears when the user types the right parenthesis.
- Complete sequences: For example, logging onto a network requires that the user perform a sequence of actions. When the user does this for the first time, the system can store the information and henceforth allow the user to trigger the sequence with a single action. The user is then not required to memorise the complete sequence.
- Correct commands: To help users to type commands correctly a system can, for example, employ command completion which will display complete alternatives as soon as the user has typed the first few letters of a command.

3.3.6.4 Design Standards

Standards concern prescribed ways of discussing, presenting or doing something. The aim is to achieve consistency across products of the same type. We are familiar with standards in many walks of life – standard colours for electrical wiring, standard shoe and clothing sizes, etc. Standards have been produced for many areas of the computer industry – paper sizes, transmission protocols, compilers, character set representations, etc. For the last decade or so, there has been a move to introduce standards into interface design.

Standards for interactive system design are usually set by national or international bodies to ensure compliance with a set of design rules by a large community. Standards can apply specifically to either the hardware or the software used to build the interactive system.

Standardization in interface design offers the following benefits:

- Provides a common terminology, so that designers know that they are discussing the same concept.
- Facilitates program maintenance and allows for additional facilities to be added.
- Gives similar systems the same 'look and feel' so that elements are easily recognisable.
- Reduces training needs because knowledge can be transferred between standardized systems.
- Promotes health and safety of users who will be less likely to experience stress or surprise due to unexpected system behaviour.

On the other hand, a user interface design rule that is rigidly applied without taking the target user's skills, psychological and physical characteristics or preferences into account, may reduce a product's usability. Standards must therefore always be used together with more general interface design principles such as those proposed by Dix et al., Preece et al. and Shneiderman discussed above.

3.4 Conclusion

In this unit we looked at some of the things designers of system interfaces do wrong, but we focused mostly on how to design correctly. In doing this we gave an overview of some of the most prominent sets of guidelines and principles for interface design

It is important to realise that design guidelines do not provide recipes for designing successful systems.

They can only provide guidance and do not guarantee optimum usability. Even when armed with very good guidelines, a designer should still make an effort to understand the technology and the tasks involved, the relevant psychological characteristics of the intended users, and what usability means in the context of the particular product.

Guidelines can help designers to identify good and bad options for the interface. They also restrict the range of techniques can be used and still 'conform' to a particular style, but they can be very difficult to apply. In many ways, they are only as good as the person who uses them. This is a critical point, because many companies view guidelines as a panacea. The way to improve an interface is not just to draft a set of rules about how many menu items to use or what colours make good backgrounds. We cannot emphasise enough that users' tasks and basic psychological characteristics must be taken into account. Unless you understand these factors, guidelines will be applied incorrectly.

NOTES

NOTES

UNIT 4
Interaction Design

CONTENTS

UNIT 4 - OUTCOMES

After studying this unit you should:

- Know and be able to describe the different interface types that current computer systems can have.
- Understand and be able to give examples of specific interaction design techniques, namely:
    - Low-fidelity prototyping.
    - High fidelity prototyping.
    - Conceptual design, including interface metaphors
- Understand what evaluation of interactive systems entails.
- Outline the main characteristics of the major evaluation techniques, including their advantages and disadvantages.
- Outline the key differences between different approaches to evaluation.

4.1 Introduction

In this unit we discuss a small selection of topics that are associated with interaction design. INF3720 – the third level module on HCI – covers the topic of interaction design in detail. Here we look at the following:

- the different forms that interfaces of interactive systems can take,
- some specific interaction design techniques
- an overview of the evaluation of interactive systems.

4.2 Interface Types

Preece et al. (2007) give an overview of the different types of interfaces. Following Preece et al., we provide a brief description of eleven of these.

4.2.1 Advanced Graphical Interfaces

The term graphical user interface (GUI) refers to any interactive system that uses pictures or images to communicate information. This is an extremely wide definition. It includes keyboard-based systems that only use graphics to present data. It also includes walk-up and use systems where users only interact by selecting portions of a graphical image.

The strength of GUIs is the way they support interaction in terms of (Shneiderman, 1998):

- Visibility: Graphical displays can be used to represent complex relationships in data sets that would not, otherwise, have been apparent. This use of graphics is illustrated by the bar charts and graphs that were introduced in the section on requirements elicitation.
- Cross-cultural communication: It is important that designers exploit the greatest common denominators when developing interaction techniques. Text-based interfaces have severe limitations in a world market. Graphical interaction techniques are not so limited. In particular, ISO standards for common icons provide an international graphics language ('lingua franca').
- Impact and animation: Graphical images have a greater intuitive appeal than text-based interfaces, especially if they are animated. The use of such techniques may be beneficial in terms of the quality and quantity of information conveyed. It may also be beneficial in improving user reactions to the system itself.

Some weaknesses of GUIs are:

- Clutter: There is a tendency to clutter graphical displays with a vast array of symbols and colours. This creates perceptual problems and makes it difficult for users to extract the necessary information. Graphical images should be used with care and discretion if people are to understand the meanings associated with the symbols and pictures.
- Ambiguity: Graphical user interfaces depend upon users being able to associate some semantic information with the image. In other words, they have to know the meaning of the image. From our earlier discussion, they have to interpret the mappings. A user's ability to do this is affected by their expertise and the context in which they find the icon.
- Imprecision: There are some contexts in which graphical user interfaces simply cannot convey enough

information without textual annotation. For instance, using the picture of an expanding and shrinking bar to represent the changing speed of a car is probably not a good idea when there is an important difference between 120 and 121 kilometres per hour...

- Slow speed: Graphical presentation techniques may not be suitable if there are relatively low-bandwidth communications facilities or low quality presentation devices. The performance problems need not relate directly to the graphical processing. Network delays may delay the presentation of results and this violates the rapid criteria for direct manipulation.

Advanced graphical interfaces involve interactive animations, multimedia, virtual environments, and visualisations. Multimedia includes graphics, text, video, sound and animations that the user can interact with. It supports quick access to multiple representations of information and is well suited for training, education and entertainment. A problem with multimedia is that users tend to favour animations and video clips and easily ignore accompanying text and static diagrams.

Virtual reality and virtual environments are graphical simulations that create the illusion that the user is part of the environment. It gives user the experience of operating in 3D environments in ways that are not possible in the real world. Virtual objects can appear very true to life. Users in a virtual environment have a first-person perspective where they see the environment through their own eyes, or a third-person perspective where they see the environment through the eyes of an avatar – an artificial representations of a real person (Dix et al., 2004).

4.2.2 Web-Based Interfaces

Web-based interfaces are graphical interfaces that are located on servers connected to the Internet and are accessed by users through web browsers. Web design is restricted by the available bandwidth and the associated download time. Although in first-world countries high bandwidth is available to most people, large numbers of internet users in developing countries do not have fast internet access. Nowadays web sites can have most of the characteristics of advanced graphical interfaces, but uncluttered design and easy accessibility of the required information are still preferable to web pages filled with flashing advertisements and lots of graphics and animations. Users should always know where they are, what they can find there and where they can go next. Web design still relies heavily on the use of text.

The main advantage of web-based interaction is that it provides users with access to large volumes of information at the click of a button. Sophisticated search engines such as Google makes it easy to search for information on specific topics. Unfortunately there are also large amounts of irrelevant information to search through and, since practically anybody can load information onto the web, a lot of what is there is not trustworthy.

Another important advantage of web-based interaction is the social aspect. It allows people to connect very easily with anybody, anywhere in the world.

4.2.3 Speech Interfaces

A speech interface allows the user to talk to a system that has the capacity to interpret spoken language. It is commonly used in systems that provide specific information (e.g. flight times) or perform a specific transaction (e.g. buy a movie ticket). Technology such as speech enabled screen readers and speech operated home control systems (e.g. for switching appliances on and off) can be especially helpful to people with disabilities. Current technology allows for much more natural sounding speech than the early synthesized speech. Speech interfaces in applications for children who cannot yet read will expand the possibilities that technology can offer them.

There are some disadvantages to these kinds of interfaces:

- They are relatively difficult to develop.
- They may not be adaptable to different accents, voice pitch and speech defects (e.g. lisping).
- They may misinterpret what the user is saying.
- Voice response may appear unnatural.

ACTIVITY 4.1
DSTV Customer Services uses a speech interface on their telephone enquiry system. For this activity you have to phone them. Your aim with this phone call is to find out what packages they offer and what the cost of each package is. Their number is: 083 900 DSTV (3788). Now describe your experience with the speech interface, pointing out specific problems and good aspects of the interface.

4.2.4 Pen, Gesture and Touchscreen Interfaces

Personal digital assistants (PDAs) come with a pen for making on-screen selections, or to write or sketch freehand. Objects can also be manipulated through swiping or stroking gestures. Pen-based interfaces are also suitable for large displays. Through a process called ‘digital ink’ that uses sophisticated handwriting recognition and conversion techniques, text written on a PDA screen or tablet PC, for example, can be converted into digital text. Gesture-based input involves camera capture and computer vision to detect people’s arm and hand gestures. This makes sign language interpreting systems possible. The latest systems use sensor technologies to detect touch, bend and speed of movement. Touchscreens allow users to manipulate screen objects with their fingers. Two hands can, for example, be used to stretch an object in two different directions at the same time.

These kinds of interfaces can increase the speed and accuracy of input, and users use natural gestures to interact. They also provide options for users who may have difficulty using the mouse and keyboard. Disadvantages are that the flow of interaction may be interrupted, incorrect options may accidentally be chosen and movement and handwriting may be misinterpreted.

4.2.5 Mobile Interfaces

These are interfaces designed for handheld devices such as cell phones that are intended for use on the move. The space limitations compel designers to use buttons for multiple purposes. For example a single key on a cell phone can represent up to five characters and each is associated with a predefined number of presses of the key. Elderly users find this particularly hard to use. The limitations on screen size restrict the font size and the amount of information that can be displayed on the screen.

A number of cell phones have been designed specifically with elderly users in mind. These typically have larger buttons with larger text on the keys and a larger display that allows for the use of bigger fonts. Unfortunately many older users do not buy their own phones but receive them as gifts from children and grandchildren whose own contracts allow for an upgrade of the phone model.

4.2.6 Multimodal Interfaces

In multimodal interfaces, different ways of interacting – including touch, sight, sound and speech – are combined so that users can experience or control information in multiple different ways. Different input or output methods are used simultaneously, for example speech and gesture, or eye-gaze and gesture. Speech and touch combinations are already being used, but few other multimodal interfaces are commercially available.

They allow more flexible interaction and can support users with disabilities or very young users. There are several disadvantages: input needs to be calibrated for accurate interpretation; they are complex and difficult to implement; and they are still very expensive.

4.2.7 Shareable Interfaces

These interfaces allow more than one user to interact with the system, providing multiple (sometimes simultaneous) inputs. Tabletop environments already exist that detects touch input from multiple users at the same time. They use an array of embedded antennae that each transmits a unique signal. The users each sit on their own chair or mat which has a receiver installed. Through the user's body, a signal goes from the tabletop to the receiver that tells the computer which antenna was touched.

Shareable interfaces provide large interactional space and supports flexible group work and sharing of information. Disadvantages are that separating personal and shared workspaces requires specialised hardware and software and correct positioning at the interface. These interfaces are also expensive to develop.

4.2.8 Tangible Interfaces

Hornecker and Buur (2006) describe tangible interaction as encompassing 'a broad range of systems and interfaces relying on embodied interaction, tangible manipulation and physical representation (of data), embeddedness in real space and digitally augmenting physical spaces' (p. 437).

These interfaces use sensor-based interaction. Physical objects that contain sensors react to user input which can be in the form of speech, touch or manipulation of the object. The effect can take place in the physical object (e.g. a toy that reacts to a child's spoken commands) or in some other place (e.g. on a computer screen). Sensors are typically RFID tags which can be stickers, cards or disks that can be used to store and retrieve data through a wireless connection with a RFID transceiver (Preece et al., 2007). Tangible interfaces have been used for urban planning and storytelling technologies, and are generally good for learning, design and collaboration. Physical representations of real-life manipulable objects enable visualisation of complex plans. Physical objects and digital representations can be positioned, combined, and explored in dynamic and creative ways.

Tangible interfaces are particularly suitable for young children. Children's body movements and their ability to touch, feel and manipulate things are important for developing sensory awareness and therefore also for their general cognitive development (Antle, 2007). Tangibles can also help children develop understanding of abstract concepts, as these are often based on their understanding of spatial concepts and how they use their bodies in space (Antle, 2007).

PETS (Personal Electronic Teller of Stories) (Montemayor et al., 2000) is a tangible storytelling system that allows children to create their own interface – a soft, robotic toy. Figure 4.1 shows an example of an interface built by children. These toys provide the interface between the child and the storytelling software that is located on a computer.

Figure 4.1 An example of a PETS creature (University of Maryland HCI Lab, 2008)

Some of the problems with tangible interfaces are development cost, accurate mapping between actions and their effects, incorrect placement of digital feedback.

4.2.9 Augmented and Mixed Reality Interfaces

In an augmented reality interface virtual representations are superimposed on physical devices and objects, while in a mixed reality environment views of the real world are combined with views of a virtual environment. Mixed reality systems have been used for medical applications, where, for example, a scanned image of organs or an unborn baby is projected onto the body of the patient to help doctors to 'see' what goes on inside the body.

These interfaces may enhance perception of the real-world, and can thereby support training and education (for example, in flight simulators). However, the added information could become distracting, and users may have difficulty to distinguish between real and virtual worlds. These systems are also very expensive.

4.2.10 Wearable Interfaces

These interfaces involve input and output devices that are integrated with normal apparel, such as headgear or spectacles. They are mobile and less restrictive than desk-based technologies or even mobile technologies. They can create a sense of realism, and provide a means of immediate feedback. This immediate feedback can be especially helpful in the detection of medical conditions. Some problems are still experienced with these interfaces including that they are uncomfortable because of size and weight factors, and they are restricted by battery life.

A well-known example of a wearable interface is the sensor built into running shoes so that trainers or family can monitor the progress of a long distance runner during a race.

4.2.11 Robotic Interfaces

These are interfaces that enable users to move and steer a remote robot. Robots are computational devices that have the physical appearance and behaviours of humans or animals. They can be built to go into places too small or dangerous for humans, or for manual repetitive tasks. Domestic robots can be manipulated to help in the house and they can be especially useful for the disabled. Pet-like robots have been developed to host events or act as companion. They contain embedded sensors that detect user behaviours and respond to them.

Sony's Aibo (Figure 4.2) is a robotic dog that can perform playful behaviour, wag its tail, walk, fall down and stand up, sit and shake hands (Bartlett et al., 2004). Despite its robotic appearance, these features are enough to convince a child that it is a being with feelings, even if their attention is drawn to its robotic features.

Figure 4.2 AIBO (Hughes, 2001)

The New Scientist (24 April 2010) reports on NASA's plan to send a humanoid robot into space. The aim is to include the robot in the crew to perform mundane mechanical tasks. The robot, called Robonaut (see Figure 4.3), will go on its first trip into space in September 2010 so that researchers can check the influence of cosmic radiation and electromagnetic interference on its performance. It consists of a head and torso with highly functional arms with which to manipulate tools.

Figure 4.3 Robonaut (from http://www.tgdaily.com/space-features/48312-nasa-builds-new-generation-robot-astronaut)

ACTIVITY 4.2

Complete the following table. Do not rely only on the information given above. Try and find examples, advantages and problems not mentioned here

Interface Type	Description	Advantages	Problems	Application examples
Web-based				
Speech				
Pen, gesture, touch screen				
Mobile				
Multimodal				
Shareable				
Tangible				
Augmented and mixed reality				
Wearable				
Robotic				

4.3 Interaction Design Techniques

According to Preece, et al. (2007) the iterative interaction design process involves, firstly, to identify the users' needs and requirements; secondly, to develop alternative designs according to those requirements and then building interactive versions (prototypes) of those designs, and, finally, to evaluate the users' experience with the product.

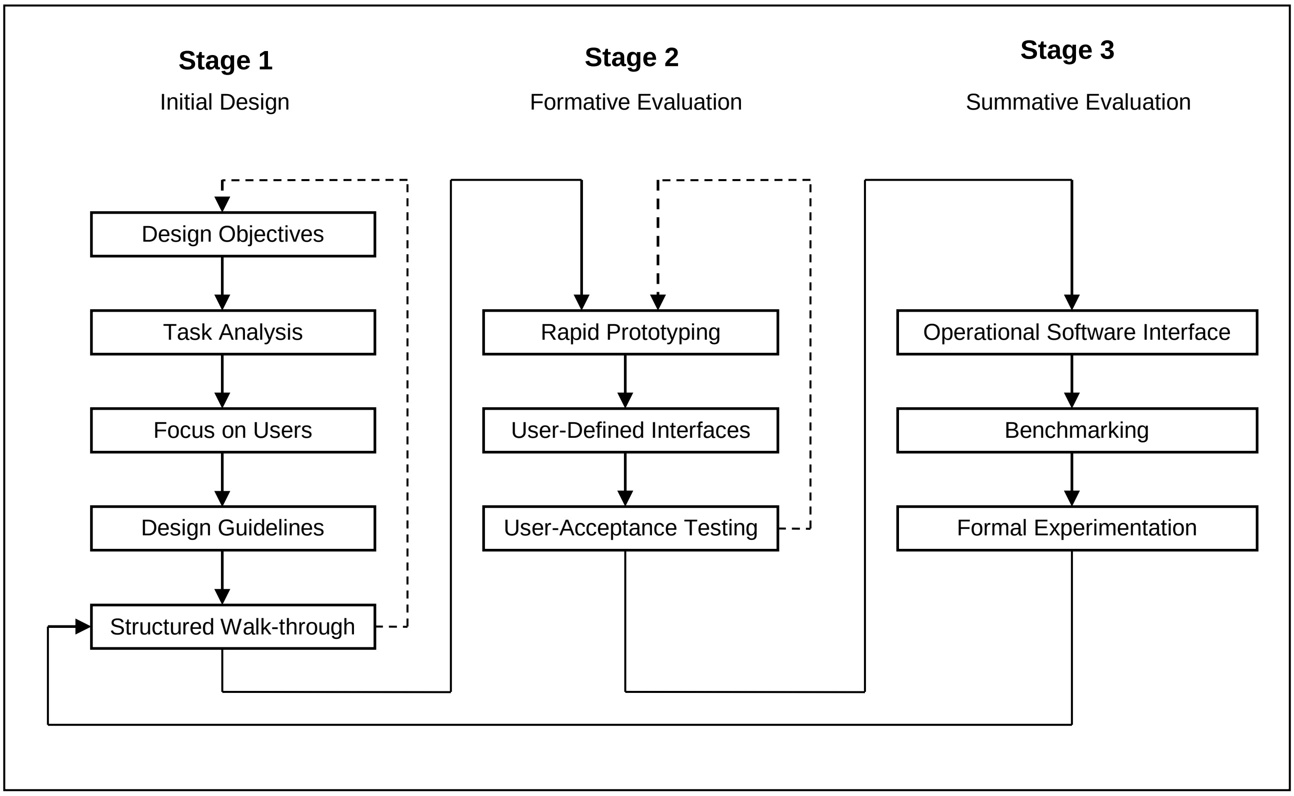


Figure 4.4 HCI Life Cycle (Williges and Williges, 1984)

More than 20 years ago Williges and Williges (1984) produced their classic model of software development whereby interface design drives the overall design process. A graphical representation of their model appears in Figure 4.3. Their standpoint is that by identifying user requirements early in the software development process, code generation and modification effort will be reduced. Not much different from what Preece, et al. (2007) are advocating.

In Units 2 and 3 we looked at some of the elements of stage 1 of this model, namely 'Focus on users' and 'Design Guidelines'. In the remainder of this unit we discuss some of the components of stages 2 and 3. Stage 2 (formative evaluation) involves prototyping and evaluation of the prototype, and stage 3 (summative evaluation) involves the evaluation of the completed system.

4.3.1 Prototypes

4.3.1.1 Definition and Purpose

Preece, et al. (2007) define a prototype as 'a limited representation of a design that allows users to interact with it and to explore its usability' (p.530). It can take the form of a simple, paper-based storyboard of the interface screens to a computer-based, functionally reduced version of the actual system.

Prototypes have several functions:

1. They provide a way to test out different design ideas.
2. They act as a communication medium within the design team – the members can test their different ideas on the prototype and the team can discuss these ideas.
3. The act as communication medium between designers and users or clients – using one or more prototypes designers can explain their own understanding of what the system should look like and what it should be able to do to users and clients. The users can then respond to that by explaining how the prototype does or does not address their needs.
4. They help designers to choose between alternative designs.

4.3.1.2 Low-Fidelity Prototypes

A low-fidelity prototype is a cheap mock-up of a system. It does not use the material that the final product will be built of and it may not even look a lot like the intended system. However, they still convey the basic components and functionality of the system and can therefore still fulfil all of the purposes described above.

The advantages of using low-fidelity prototypes are that they are cheap and can be produced very quickly. They can therefore be adapted very easily and without much cost. They are particularly useful if the designers are just beginning and still need to explore different ideas. At that point you don't want to spend too much on a sophisticated prototype just to find out you have completely missed the point with your design.

Low-fidelity prototypes are not meant to become part of the real system – they are usually thrown away when they have served their purpose (or kept for sentimental reasons).

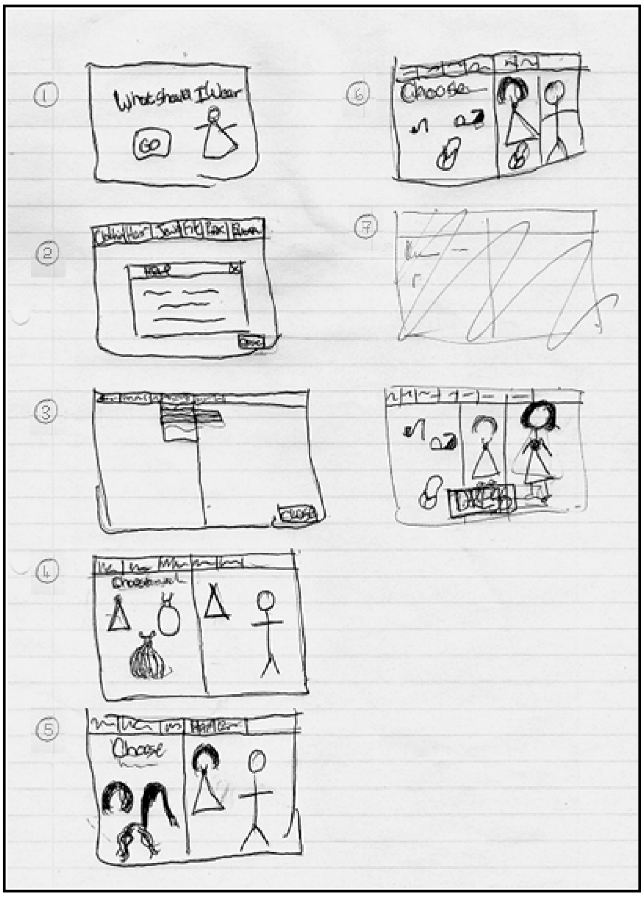


Figure 4.5 A low-fidelity prototype

Examples of low-fidelity prototyping are:

- Storyboards – a series of simple images representing the screens to show how the user will progress through an application. Figure 4.5 is such a storyboard created by a twelve-year old trying to explain her idea for a computer application that helps girls decide on an outfit for the day.
- Sketching – more detailed sketches of the interface, including what the icons will look like, etc. This requires better drawing abilities than storyboards.

- Index cards – each card represents a screen or an element of a task and the user can be taken through different sequences of these.
- Wizard of Oz – here you have a basic software-based prototype but it still lacks functionality. The user uses it as if it has all the functionality, but actually the feedback is provided but someone sitting in a different room at a computer that is connected to the user's computer. That operator can then make up responses to the user's actions as they go along.

4.3.1.3 High-fidelity Prototypes

High fidelity prototypes resemble the final system and usually use the same materials that would be used in the final product. This requires software tools and programming skills. Windows programming languages such as Delphi and Visual Basic are very powerful prototyping tools that can be used with only basic programming skills.

Contrary to low-fidelity prototyping, this method is time-consuming and can be expensive. Because they take long to build they cannot be adapted easily to changing ideas or requirements. The developers of these prototypes are also more reluctant to change them because of the effort put into them. Because the prototype looks so much like the final product the testers tend to comment on superficial aspects such as the look and feel rather than on the basic functions included. A software prototype can create high expectations and make it look as if more is possible than can actually be done. One bug in a computer-based prototype system will make it impossible to use.

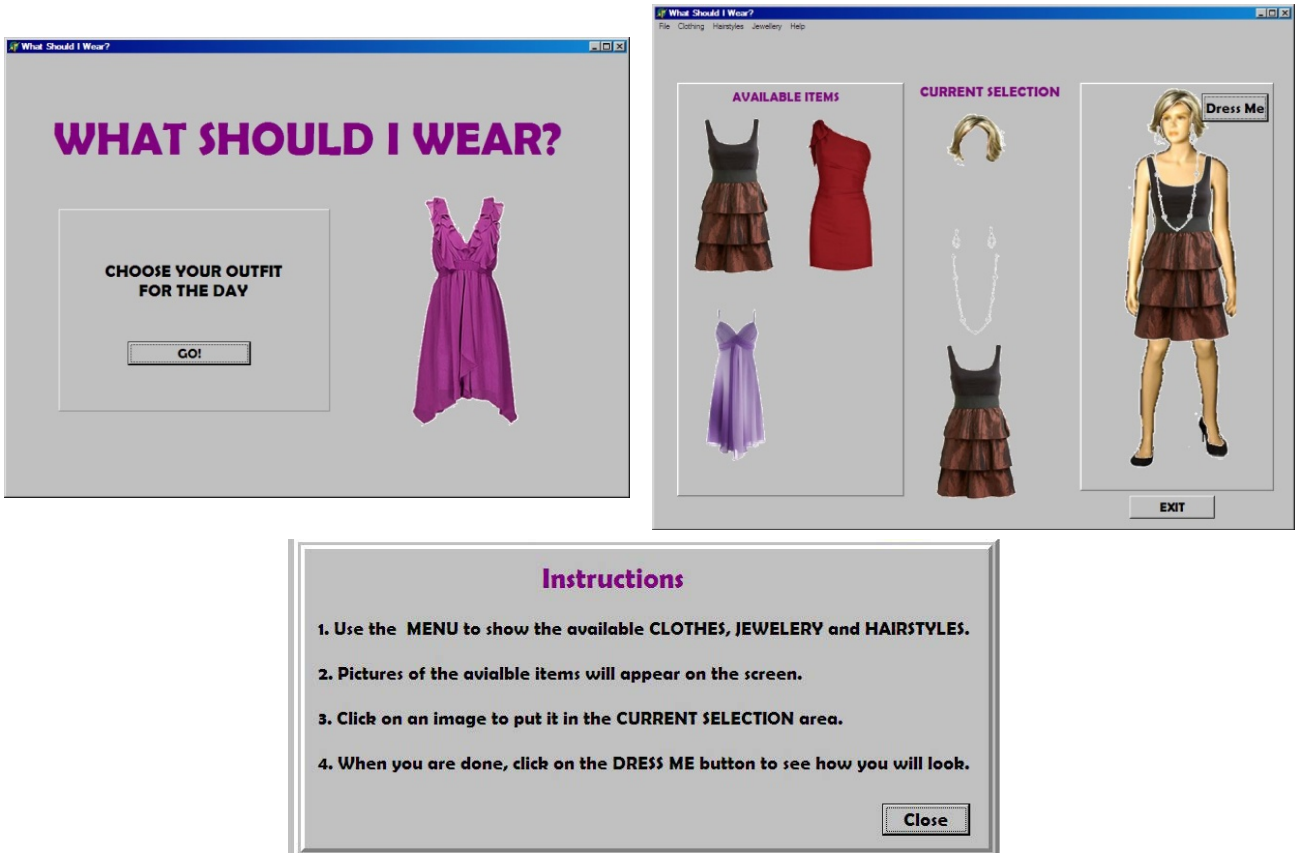


Figure 4.6 Screens from a high-fidelity prototype created by a 12-year-old using Delphi

An advantage of this kind of prototype is that it can gradually develop into the final product so the time and resources put into it can be worthwhile – this is called evolutionary prototyping.

Figure 4.6 shows a few screens from a high-fidelity version of the low-fidelity prototype shown in Figure 4.5. It was developed in Delphi by the twelve-year old. Although the functionality is not real in this prototype, to the user it appears as if it is.

ACTIVITY 4.3

Compare low and high fidelity prototyping by completing the following table.

Type of prototype	Advantages	Disadvantages
Low-fidelity		
High-fidelity		

ACTIVITY 4.4

Suppose you have been asked to design a web-based system for renting online DVDs. The system allows users to

- browse the available movies,
- select one or more that they want to watch,
- make a credit card payment,
- receive the access keys for the selected movies via sms.

The idea is that the user can then watch the movies online (within a 24-hour period) from the store's web site by providing the access keys.

Create a low-fidelity prototype (in the form of a storyboard) for this system.

4.3.2 Conceptual Design

Conceptual design involves turning the users' needs and requirements into a conceptual design (Preece et al., 2007). Preece et al. define a conceptual model as a 'high-level description of how a system is organised and operates' (p. 51). It is not a description of the interface – it gives an idea of what users can do with a system and what concepts they need to be familiar with in order to use it. Using prototypes is one way of getting the conceptual model of the intended system right.

Preece, et al. provide the following principles to follow when doing the conceptual design, namely:

- Keep an open mind but always think of the users and their context.
- Discuss the design ideas with all stakeholders as often as possible.
- Use low-fidelity prototyping to get quick feedback.
- Continue doing the above over and over until you are sure you have the correct conceptual design.

The conceptual design process requires that the designer determine how the functions to be performed will be divided between the system and the users; how these functions relate to each other and what information should be available to perform these functions (Preece et al., 2007).
Two important factors in conceptual design are interface metaphors and the interface type.

4.3.2.1 Interface Metaphors

An interface metaphor is an important component of a conceptual model. It provides a structure that is similar to some aspects of a familiar entity, but that also has it own behaviours and properties. We use them to explain something that is hard to grasp by comparing it with something that is familiar and easy to grasp. The Windows desktop is a familiar interface metaphor – the computer screen is like a desktop and the folders and applications are like the things we could in real-life have on top of a desk.

Figure 4.7 Using a 'conquer the ogres' metaphor to teach the multiplication tables

The metaphor that is chosen for an interface must fit the task and it must be suitable for the intended users. Metaphors that appeal to adults may have no meaning for children. The children's game, Storybook Weaver, uses a book metaphor allowing children to create a cover page and then to fill the pages of the book one by one. Metaphors can also be used to turn something that is potentially boring into an engaging experience. Most children hate learning the multiplication tables. Timez Attack is a children's game that uses a 'conquer the ogres' interface metaphor to make this an exciting activity. Figure 4.7 shows a screen from this game: the ogre presents the multiplication sum and the child (represented by a little green alien) has to build and present the answer within a restricted period of time.

The purpose of a metaphor is to provide a familiar structure for interaction. The metaphor must be appropriate for the task. The user should not be in a position to apply aspects of the metaphor to the system if these aspects are not applicable or will lead to confusion.

ACTIVITY 4.5

1. Identify any interface metaphor in a system you are familiar with. Answer the following questions about this metaphor:
   a. Does it supply structure to the interaction process? How?
   b. How much of the metaphor is relevant to the use of the system? (In other words, are all aspects of the metaphor relevant or only some? Can those that are not relevant lead to confusion in the interface?)
   c. Is the metaphor easy to interpret? (Will users easily understand the metaphor?)
   d. How extensible is the metaphor? (If the application is extended, will unused aspects of the metaphor be applicable?)

2. Identify and describe a suitable interface metaphor for the online DVD renting application of Activity 4.4.

4.3.2.2 Interface Types

When doing the conceptual design, designers should preferably not be influenced by a specific, predetermined interface type. Having a specific interface type in mind may stifle the design process and potentially good solutions may be overlooked. Another approach would be to reinterpret an initial conceptual design for all the different types of interfaces (in section 4.2 we described eleven) and consider the effect that the change in interface type has on the design. Any system comes with constraints on the type of interface that can be used.

4.4 Evaluation of Interactive Systems

In the third level HCI module – INF3720 – evaluation will be dealt with in detail. In this module we only provide and overview of the process and the methods involved.

Evaluation is a key aspect of human-computer interaction (and of interaction design in particular). It refers to the validation of an interactive system against human-computer interaction requirements (Dix et al., 2004). Any design should be assessed and all systems tested to ensure that they meet the users' requirements. Evaluation is not a single phase that comes at the end of the design process, but rather an activity that is used throughout the design process to provide feedback on the design right from the beginning. For Dix et al. (2004) the three main goals of evaluation are then to assess the extent of the system's functionality, to assess the effect of the interface on the user and to identify specific problems with the system.

Dix et al. (2004) and Williges and Williges (1984) (see Figure 4.4) distinguish between formative and summative evaluation.

4.4.1 Formative and Summative Evaluation

Formative evaluation is done early in the design process and continues through the design cycle to support design decisions (Dix et al., 2004). Low cost techniques such as low-fidelity prototyping are suitable for formative evaluation. Early evaluation helps to predict the usability of a product and assesses the designer's understanding of the user requirements.

There are many times during design when designers need answers to questions in order to check that their ideas really are what users need or want. In a sense, evaluation meshes closely with design and guides the design by providing feedback. If formative evaluation is to guide development, then it must be conducted at regular intervals during the design cycle.

Formative evaluation is used to prevent problems when users start to operate new systems.

Summative evaluation is done at the end of the design cycle and tests the end product (Dix et al., 2004). Its aim is to demonstrate that the completed system fulfils its requirements or to identify problems users have with the system. Usability testing with real users is suitable for summative evaluation. Whereas formative evaluation tends to be exploratory, summative evaluation is often focussed upon one or two major issues. Interaction designers will be anxious to demonstrate that their systems meet company and international standards as well as the full contractual requirements.

The bottom line for summative evaluation should be to demonstrate that people can actually use the system in their working setting. If sufficient formative evaluation has been performed, then this may be a trivial task. If not, then it becomes a critical stage in development.

We can summarise as follows how evaluation fits into the design life cycle:

During the early design stages, evaluation tends to be done to:

- Predict the usability of a product or an aspect of it.
- Check the design team's understanding of user requirements by seeing how an existing system is being used in the field.
- Test ideas quickly and informally as part of envisioning a possible design.

Later in the design process the focus shifts to:

- Identifying user difficulties, so that the product can be more finely tuned to meet their needs.
- Improving an upgrade of the product.

4.4.2 How to Evaluate

Evaluation can be done in laboratories or in the real-life environment where the system will be used. Evaluation methods and the evaluation setting is closely linked. Preece et al. (2007) identified three main evaluation approaches which we discuss in brief.

4.4.2.1 Usability Testing

Usability laboratories with sophisticated audio and video recording facilities, specialised hardware and software for recording and analysing users' behaviour when using a system, are often used for usability testing. Such a laboratory setting gives the evaluator control over the conditions of the study, but it removes the natural context (with associated interferences) which may be important in the use of the system.

During usability testing, typical users perform selected tasks while their actions are recorded. The evaluator analyses the data collected to judge performance, identify errors and explain user behaviour. Evaluators should not directly interact with the user in a way that can skew the results. The intention is to derive some measurable observations that can be analysed using statistical techniques. For this approach to be successful, it requires specialist skills in HCI.

Usability experiments are usually supplemented with interviews and satisfaction questionnaires.

4.4.2.2 Field Studies

This type of evaluation is done in natural settings. The aim is to understand what users do naturally and how the technology affects them in the real-life environment. The evaluator can be an outsider that observes and records what is happening, or an insider or participant that enters the world of the user to experience the impact of the technology first-hand. Evaluation done in the real environment of use provides the natural context of use but it may be more difficult to set up the required equipment. Subjects can still be influenced by the presence of researchers in their working environment.

This technique avoids the problem of alienation or irritation that can sometimes be created by the use of interviews and questionnaires. The problem with this approach is that it requires a considerable amount of skill. To enter a working context, observe working practices, and yet not affect users' tasks seem to be an impossible aim.

4.4.2.3 Analytical Evaluation

This either a heuristic evaluation, that involves experts who use heuristics and their knowledge of typical users to predict usability problems, or walkthroughs where experts 'walk through' typical tasks. The users need not be present and prototypes can be used in the evaluation. Popular heuristics such as that of Nielsen (2001) were designed for screen-based applications and are inappropriate for technologies such as mobiles and computerised toys.

There are circumstances where combinations of the three techniques will be appropriate. Other evaluation techniques that can be combined with the three methods discussed above, or that can be performed as part of these methods, are:

4.4.2.4 Cooperative Evaluation Techniques

Cooperative evaluation techniques are particularly useful during the formative stages of design. They are less clearly hypothesis-driven and are an extremely good means of eliciting user feedback on partial implementations.

The approach is extremely simple. The evaluator sits with the user while they work their way through a series of tasks. This can occur in the working context or in a quiet room away from the ‘shop-floor’. Designers can either use low-fidelity prototyping or partial implementations of the final interface. The evaluator is free to talk to the user as they work on the tasks but it is obviously important that they should not be too much of a distraction. If the user requires help, then the designer should offer it and note down the context in which the problem arose. The main point about this exercise is that subjects should vocalise their thoughts as they work with the system. This can seem strange at first, but users quickly adapt. It is important that records are kept of these observations, either by keeping notes or by recording the sessions for later analysis.

This low cost technique is very good for providing rough and ready feedback. Users feel directly involved in the development process. This often contrasts with the more experimental approaches where users feel constrained by the rules of testing.

A limitation of cooperative evaluation is that it provides qualitative feedback and not measurable results. In other words, the process produces opinions and not numbers. Cooperative evaluation is extremely ineffective if designers are unaware of the political and other pressures that might bias (that is, positively or negatively influence) a user’s responses.

4.4.2.5 Scenario-Based Evaluation

Scenarios are informal narrative descriptions of possible situations. In interface design it is a sample trace of interaction. This approach forces designers to identify key tasks in the requirements’ gathering stage of design. As design progresses, these tasks are used to form a case book (containing standard tests) against which any potential interface is assessed. Evaluation continues by showing the user what it would be like to complete these standard tests using each of the interfaces. Typically, they are asked to comment on the proposed design in an informal way. This can be done by presenting them with sketches or simple mock-ups of the final system.

The benefit of scenarios is that different design options can be evaluated against a common test suite. Users are then in a good position to provide focussed feedback about the use of the system to perform critical tasks. Direct comparisons can be made between the alternative designs. Scenarios also have the advantage that they help to identify and test design ideas early on.

The problems with this approach are that it can focus designers’ attention upon a small selection of tasks. Some application functionality may remain untested, while users become all too familiar with a small set of examples. A further limitation is that it is difficult to derive measurable data from the use of scenario-based techniques. In order to do this, they must be used in conjunction with other approaches such as usability testing.

4.4.2.6 Query techniques

The two main types of query techniques are interviews and questionnaires.

Interviews

Using interviews for evaluation purposes involves asking users questions about their experiences with the system being evaluated. For an interview to be effective, the interviewer will plan it carefully, preparing specific questions or making a list of topics to address. This will help to focus the interview and to keep the issues discussed with different users consistent. However, the interview should not be overly structured, so that the interviewer can easily adapt the questioning to suit the specific user. Structured interviews are easier to conduct and analyse than flexible interviews. On the other hand, they may miss important details relating to the user's experience which a more flexible interview would have picked up. A good compromise is a semi-structured interview which is based on leading questions, but has the flexibility to investigate promising or unanticipated directions.

In general, the advantages of interviews are that the evaluator can vary the level of questioning to suit the context, and that the evaluator can probe the user for more information on relevant issues which arise spontaneously during the interview.

Questionnaires

A questionnaire consists of fixed questions relating to interaction with the system being evaluated. When used for the purpose of evaluation, users complete the questionnaire after completing a given task or set of tasks. This technique is less flexible than interviews but it has several advantages. A larger number of users can be included in the evaluation as it is less time-consuming and labour intensive and the results can be analysed more rigorously. Different styles of questions can be used, namely open-ended questions (which allows the user to freely give his or her opinion) as well as closed questions such as multiple choice or ranked questions. Questionnaires can be completed in fixed sessions, but they can also be administered independent of time and place, and without the presence of an evaluator.

4.4.2.7 Heuristic evaluation

This evaluation technique was developed by Jakob Nielsen and his colleagues (Nielsen, 1994). Applying heuristic evaluation means that user interface design experts evaluate the user interface according to usability principles known as heuristics. The process of heuristic evaluation involves three steps:

1. Briefing: experts are told what to do.
2. Evaluation: each expert spends a few hours taking at least two passes through the interface, using the heuristics to identify problems.
3. Debriefing: experts meet to discuss their evaluations, prioritize problems and suggest solutions.

The advantage of heuristic evaluation is that there are fewer practical and ethical issues to take into account as users are not involved. A disadvantage is that the experts often identify problems that aren't really problems. This suggests that heuristic evaluation should preferably be used along with other techniques and that several evaluators should take part in the evaluation.

We end this section with a list of Nielsen's evaluation heuristics formulated as questions:

1. How good is the visibility of system status?
2. Is there a clear match between the system and the real world?
3. Does the user have control when needed and are they free to explore when necessary?
4. Does the user interface display consistency and adherence to standards?
5. Does the interface help users recognise, diagnose, and recover from errors?
6. How good is the error prevention?
7. Does the interface rely on recognition rather than on recall?
8. How flexibility and efficient is it to use?
9. How good is the interface in terms of aesthetics and minimalist (clear and simple) design?
10. Is there adequate help and documentation available?

ACTIVITY 4.6

Use Nielsen's heuristics to evaluate the assignment submission pages of the myUnisa web site. Your evaluation report should provide answers to each of the ten questions and should include examples of interface elements where applicable.

4.5 Conclusion

This unit provided a bit of a mixed-bag of information on interaction design. We described a range of different interface types that are currently available. We also looked at techniques that designers should use during the interaction design process – specifically, prototyping and conceptual design.

Evaluation plays a major role in interaction design. Without doing some form of evaluation it is impossible to know whether the system fulfils the needs of the users and how well it fits the physical, social and organisational context in which it will be used. This unit provided an introduction to evaluation methods that can be used in the design of interactive systems.

NOTES

UNIT 5

Social Aspects of Computer Use

CONTENTS

UNIT 5 - OUTCOMES

After studying this unit you should:

- Understand the impact of computer technologies on society, and specifically on
  - E-commerce and business
  - Our working lives
  - Education
  - Information processing
  - Privacy and security.
- Be able to identify the problems for society that are associated with advanced computer technology.
- Be able to demonstrate a more detailed understanding of the impact of social networking web sites – their advantages and disadvantages.
- Be able to define the 'digital divide' and describe its causes and how it can be addressed.

5.1 Introduction

Our world and all aspects of our lives have become inundated with computer technologies. On the one hand we are empowered by these technologies and they improve our quality of life, but, on the other hand, they pose threats such as invasion of our privacy and widening the disparity between the rich and the poor. This last unit of the study guide introduces students to the wider social implications of computer technology.

5.2 The Impact of Information Technology on Society

In this discussion of the ways in which computers, and technology in general, influence how we live and relate to our environment will focus on the following aspects:

- E-commerce and business
- Our working lives
- Education
- Information processing
- Problems associated with advanced computer technology.

5.2.1 Business and E-commerce

Computer technology and the Internet in particular have completely changed the way commercial companies do business and how banks function. Physical distance no longer puts a restriction on the way businesses are structured. A company can have divisions located in different countries and responsibilities can be divided between countries on the basis of the skill and expertise located there. By dividing labour across countries, production companies can now have people working 24 hours of the day. Manufacturing can be allocated to where it can be done most efficiently.

Network infrastructure and online availability of services and information have made sales clerks, stock brokers and travel agents redundant. Before, they were the main information links between companies and clients but web-based services have taken over this function.

The fact that products such as software and music can be ‘shipped’ electronically has reduced the need for distribution and shipping companies. Looking at the music business as an example, digital music has completely changed the functioning of the music industry world-wide. From the consumer’s perspective this is all positive as music lovers can now buy all their music on the web. The idea that any album is hard to come by does not exist anymore. Every obscure, ‘indie’ band now sell their music online. Another advantage is that you can buy individual songs – there is no need to buy the whole album if you are only after one or two tracks. From the record companies’ and producers’ point of view this change has forced them to rethink their business models. There is now much more competition – any artists can easily market and sell their music on the Internet without the help of a record company. Advances in computer technology have also impacted the recording and production of music. High quality recordings can now be made using mobile studios and complete sound mixing and editing systems now come in the form of software applications (see Figure 5.1). The cost of making an album if you have the technological skills is much reduced.

Figure 5.1 A desktop sound mixer (left) and Apple's Garageband screen-based mixing interface

Besides the exclusion of shipping and distribution, there are numerous ways in which replacing a physical business with an online one reduces cost:

- No physical store needs to be set up and maintained.
- Simplification of order placement and execution.
- Providing 24 hour customer support.
- Staffing requirements are reduced.
- A retail business does not need to carry the inventory of a physical store.
- No restriction on retail hours.

The main cost now lies with the setting up and maintenance of the store web site, but since it is always 'open' and can be reached by millions of potential customers, spending on the usability and appearance of the site is justified.

Shipping costs of large and expensive products bought over long physical distances can increase the cost, but still, these products may not be available to the customer in any other way.

Unfortunately e-commerce creates opportunities for fraud and theft. Measures to prevent this and insurance against it may add cost to the business. It is quite easy to get unlawful access to digital music on the Internet. The same applies to the movie business. According the New Scientists (13 March 2010) there were 7.9 million pirated downloads of the movie 'The Hurt Locker' before it won six Oscar awards in 2010. In 2009, 33% of all Internet users in the United Kingdom were using unofficial online sources of music (New Scientist, 5 December 2009).

5.2.2 Working Life

5.2.2.1 Communication and Groupware

The most pronounced changes that technology has brought into the workplace are the electronic mechanisms for communication such as e-mail and Skype that allow workers to correspond cheaply and instantly over long distances. Collaborative work can be done by people who reside in different countries and who might never meet face to face.

Web 2.0 technology is commonly used by organisations to support collaborative work (this utilisation of Web 2.0 within a secure environment developed into what is sometimes called Enterprise 2.0). The historical term used for collaboration through computer technology is Computer Supported Cooperative Work (CSCW). CSCW is concerned with the principles according to which computer technology support communication and group work. The physical systems through which CSCW manifests are collectively referred to as Groupware. These systems never gained widespread popularity due to technical and design issues such as hardware and operating system incompatibilities and the inability to understand the effects of how groups and organizations function. Some of the specific problems are:

- Synchronous and asynchronous systems: It may be difficult for users to know exactly who else is using the system. Synchronous means ‘at the same time’. Asynchronous means ‘at different times’ or independent. You get both types of CSCW systems. For example, a system that supports collaboration between groups in Australia and South Africa will be asynchronous. Time differences mean that there would only be small periods of time when they would both be working on the system. If the application is asynchronous then many of the problems of contention (see below) do not arise.
- Contention: This occurs when two or more users want to gain access to a resource that cannot be shared. For example, it may not be possible for people to work on exactly the same piece of text at the same time.
- Interference: This arises when one user frustrates another by getting in their way. For example, one person might want to move a piece of text while another attempts to edit it. Similarly, one user might want the others in a group to vote on a decision while another user might want to continue the discussion.

5.2.2.2 Access

Easy access to information has also impacted on the work environment. Employees are empowered by the electronic availability of company reports and policies on internal networks. Further empowerment comes with e-mail that makes it easier for employees at lower levels to communicate with their superiors. The problems of intimidating face-to-face encounters are reduced and people can communicate with superiors without the fear of intrusion (as with personal meetings and telephone calls). In other words, managers have become more accessible.

5.2.2.3 Office Hours and Location

Technology also affects the way companies think about office hours and office space. Mobile technology allows people to do their work anywhere, any time and a centralised office may not be important any longer. This will benefit companies who can cut down on office space and it will be advantageous to employees who will have more flexible work hours. People do not need to live close to the office.

On the down side, it has become difficult to separate one’s personal and working lives. Being connected at all times heightens the need for skills such as prioritising, focusing and working without interruption.

5.2.3 Education

As UNISA students you have firsthand experience of how advances in technology have influenced education. A learning management system such as myUnisa makes it possible for students to communicate with their lecturers, participate in discussion with their fellow students, submit their assignments (at the last minute), check their examination and assignment schedules, and more. Lecturers can send urgent notices to students via SMS.

You can download an electronic copy of this study guide from the INF1520 tutorial matter page, store it on your mobile phone, iBook or Notebook and read it while lying next to the pool.

There is a vast amount of educational resources available on the Internet. Renowned universities, such as MIT in Boston, make courseware available on the internet for free (see the MITOpenCourseware site at http://ocw.mit.edu/index.htm).

ACTIVITY 5.1

David Kieras is a world leading researcher in HCI, specialising in the psychological aspects of the field. He is a Professor in the Electrical Engineering and Computer Science Department at the University of Michigan. The Internet makes it possible today to 'attend' lectures by academics of his stature from anywhere in the world.

If you have access to a fast internet connection, go to the following web site http://videolectures.net/chi08_kieras_phc/. Watch the video of the lecture titled Psychology in Human-Computer Interaction by David Kieras.

List five important aspects of HCI that you have learnt from this lecture. (You will need this list in the next activity.)

Unfortunately there is probably more low quality, unreliable information than there are trustworthy academic sources. Children should therefore be taught from an early age to distinguish between good information and information that is questionable.

A current hot topic in HCI research is m-learning (mobile learning) that involves the use of cell phones and other mobile devices as a delivery mechanism in education. Almost all students today own or have access to a cell phone, while many still do not have easy access to computers. Cell phones are therefore an ideal platform to distribute learning material to students. It does however introduce a design challenge – how does one present the material on such a small display?

There are problems associated with these trends in education. The younger generations (i.e. the students) may expect more from technology than what the older generation (the lecturers) feel comfortable with or may think is possible to do. Another has to do with the digital divide discussed in section 5.3 below – not everybody has access to the technological resources required for e-learning and m-learning. Some of the activities in this unit, for example, require access to fast internet connections and sufficient bandwidth. Not all students will have this.

ACTIVITY 5.2

Go to the INF1520 discussion forum on myUnisa and do the following:

1. Post a message with at least one suggestion on how to improve the teaching of this module using technology. Use 'Activity 5.2' as the subject.

2. Post a message with the list of five things you learnt from the Kieras lecture (see Activity 5.1). Use 'The Kieras lecture' as the subject. Some students may not have been able to do Activity 5.1 due to bandwidth problems. If those students who were able to watch the lecture post their insights on the INF1520 discussion forum, then the students who could not watch it will also get the benefit of that activity.

5.2.4 Information Processing

The processing power of supercomputers has made it possible to process huge amounts of data in relatively short periods of time. It is now possible to develop computer models of complex systems. Climate modelling is one field that relies on the computing power available today. This involves numerical representation of aspects of the climate system that makes it possible to predict future weather patterns. Just a few years ago it was difficult to simulate weather conditions of extended time periods. Now, by clustering high-speed processors to perform the calculations simultaneously, a simulation of a full year's weather takes under three weeks to complete (Lepage et al., 2006).

The human genome project provides another example of research that required this kind of computing power. Without this capacity the aim of establishing a human DNA sequence could not have been reached.

The advantages for society of being able to model complex systems include (Muglia, 2010):

- Improved the understanding of pandemics, contagion and global health trends.
- Better prediction of the impact of climate change on the environment, the economy and on humans in general.
- Better prediction of natural disasters and their impact so that effective response plans can be set up.

Google Earth (earth.google.com) enables us to 'visit' places that are not physically possible. The system's viewing interface allows users to get a bird's eye view of just about any location in the world. The system uses a search engine, satellite images, maps, 3D buildings, etc. One can zoom in to get close to a specific building or other location, and rotate the view to change the 3D perspective. Using Google Earth requires high bandwidth and a computer with relatively high processing power.

ACTIVITY 5.3

Only students who have fast internet access will be able to do this activity. You need to access Google Earth (http://earth.google.com) and download and install the software if you do not already have access to it.

Suppose you are offered a scholarship to work in the USA at the University of Maryland, College Park for a year. Your spouse and two children will accompany you. You decide that you want to live relatively close to the university, but definitely close to a primary and a secondary school that will be suitable for your children.

Use Google Earth to find a good location where you can rent a home. Identify a specific address.

5.2.5 Problems Associated with the 'Information Age'

The 'myth of the cyborg' has had an important impact upon popular attitudes towards the safety of technology. This myth has been popularised by Hollywood, in films such as the Terminator series. It is, however, part of a longer stream of literature. These earlier sources include writers such as H.G. Wells and Mary Shelley. They were less concerned about the nature of the technology itself than in the (potentially subservient) relationship between people and the machines that they create. In modern times, the introduction of complex, computer-controlled systems has increased the relevance of this relationship. However, the threat may be less spectacular than that portrayed by Terminator. We are not threatened today by a malign supercomputer seeking to destroy humanity. However, our everyday safety is almost entirely dependent upon information technology. There is a very real sense in which we are all cyborgs. We are all dependant on machines to survive.

In this section we briefly look at three specific problems that can be linked to the almost immeasurable availability of information, namely the problem of keeping information private and secure, the difficulty of filtering reliable information, and our dependence on technology.

5.2.5.1 Privacy and Security Issues

ACTIVITY 5.4

Think carefully of the places on the Internet and other networks (for example, mobile networks, electronic banking data and Facebook) where information on you is possibly stored. Make a list of everything a clever hacker can find out about you just by accessing these data sources.

The increase in computing and communications power poses a threat to both the public and the private sectors. Government, banks and private institutions keep electronic data on individuals and it is not always clear how safe that information is. Many people aren't even aware of the fact that information about them is kept in databases that may not be completely secure. If you are a DSTV subscriber, your decoder or PVR is connected to a network through which information such as your daily viewing patterns and what programmes you record can be accessed. The records of calls and SMS messages of cell phone users are accessible through the mobile network they are subscribed to. The kind information people make available of themselves on social network sites such Facebook is an obvious indication that they are not concerned about the fact this information can be accessed by people with intentions other than the subscriber had in mind.

People regard digital data as different from physical information. Because it is not tangible, they think it is less important to secure it. This is a dangerous assumption. The real difference between digital and physical sources of information is that it is easier to copy and forge digital data. It is also easy to search for any information that the user might leave unprotected.

An annoying breech of e-mail users' privacy comes in the form of spam – unsolicited mass mail that is sent to millions of users daily. New Scientist (27 February 2010) reports on a study done over one month in 2008 regarding pharmaceutical spam. The following was found: For 35 million spam messages sent, only 8.2 million reached a mail server. Of those, only 10 500 recipients clicked the link in the mail and 28 actually bought products. The 35 million messages made up only 1.5% of what one spam botnet (a network of remotely controlled computers) produced in a month. Extrapolating from this information, that specific spam botnet generated around $3.5 million in pharmaceutical sales in 2008. So, although only few people respond to spam messages, it is still worth the trouble for the spammers.

The increasing value of the information being stored and transferred across the world's computer networks is also increasing the importance of security. In the past, organisations could preserve their security by denying all external access to their systems. The increasing use of the web to advertise and sell products has meant, however, that more commercial systems are hooking up to the Internet. The increasing communication opportunities provided by electronic mail and social networks have also encouraged greater inter-connection. All these factors increase the stakes for malicious and criminal users. Electronic fund transfers and commercially sensitive e-mail messages are tempting targets.

The technological sophistication of the general population is increasing. This means that more and more people have the knowledge and ability to 'beat the system' and that commercial and governmental organisations must continually stay one step ahead of the people who 'hack' or 'crack' into their systems.

Software that is developed for the sole purpose of doing harm or gaining unlawful access to information is referred to as 'malware'. This generic term includes intrusive code such as:

- Trojan horses: This form of attack is named after the way the Greeks attacked Troy by hiding inside a 'gift horse'. In computer terms, a malicious piece of code is hidden inside a program that appears to offer other facilities. For example, a file named really_exciting_game will provide a boring game but also contain a program that attempts to access your password file. Once the program has obtained a list of user names and passwords, it may write them to a file that is visible to the attacker. From then on, the gates are open and the system is insecure. An alternative approach would be for the program to continue to run after a user thinks he/she has quit. The intruder might then be able to use the still running program to gain access to your files and resources.

- Time bombs: These are, typically, left as a means of retaliating when an employee is dismissed. For example, a program might be scheduled to run once every month. The code would check payroll records to see if an employee's name is on it. If this is the case then nothing else happens. If, however, the name is no longer there, then the program takes some malicious action. For example, it might delete the rest of the payroll or move money to another account. Such programs indicate major security breech, because they require access to personnel data and they require the ability to take some malicious action. The long term consequence may, however, be less severe than a Trojan Horse, because the system may be left secure in spite of the damage caused.
- Worms: These are self-replicating programs. These represent a major threat because they will gradually consume more and more of your resources. Your system will slowly grind to a halt and all useful work will be squeezed out. This useful work will include attempts to halt the growth of the worm. It takes considerable technological sophistication to write a worm. You must first gain a foothold in the target machine. From there you have to create a copy that can be compiled and executed on that host. This copy must then move on to generate a further copy, and so on. The main difference between a virus and a worm is that a worm does not need a host to cause harm.

The producers of malware are referred to by names such as black hats, hackers, and crackers. (Note: some legitimate programmers call themselves 'hackers' because they hack the code into shape. The media often refer to people who attack computer systems as 'hackers'. Many programmers find this annoying and would prefer the term 'crackers' for people who attack systems.)

It is generally assumed that most security violations within large organisations come from within that organisation, either through malicious actions or through carelessness. At one extreme this takes the form of industrial and military espionage. At the other extreme, security may be breeched as the result of someone leaving a flash drive or print-outs in a public place.

As the greatest security threats come from within an organisation, it follows that many companies have clear rules of disclosure. These specify what can and what cannot be revealed to outside organisations. These rules extend to the sort of access that may be granted to the company's computer systems. A particular concern here is what repair facilities may be provided for machines that contain sensitive data. One of the most effective means of breeching security is to act as a repair technician and copy the disks of any machine that you are working on. Finally, security is based around a transitive closure of the people that you trust. This basically means that if you pass information onto someone you trust, then you'd better be sure that you trust all of the people that they trust and so on. If they pass your information on to someone else, then you have to trust all of the people that this new person trusts as well.

5.2.5.2 Information Overload

New information leads to new invention, and consequently, contributes to the evolution of humankind. Our existence depends on knowledge, so we are naturally predisposed to crave information. The Internet has taken this to a different level – we now have more information to our disposal than is good for us.

People are spending large amounts of time searching through and taking in irrelevant or useless information just because it is there (Konsbruck, n.d.). Much of what appears on the Internet is incomplete, unsubstantiated and incorrect. Even worse, people now have access to harmful information such as:

- Instructions to build explosive devices.
- The genome of the influenza virus that killed over 50 million people in 1918 (Parsons, 2010). (With the right tools anyone can re-construct it).
- Political propaganda.
- Pornographic and violent video material.

There is a need for research to determine how people judge the credibility of information and systems need to be developed to help people survive the information overload (Konsbruck, n.d.). Being able to filter information is a skill that has become extremely important and mechanisms have to be developed to help those who struggle with this.

5.2.5.3 Dependence on Technology

Modern society is almost entirely supported by information technology. Although there are individuals and even communities that still function without access to technology infrastructure, society has, for the most part, reached a point of complete dependence on technology. The risk of this dependence is that the breakdown of technological infrastructure will lead to serious disruption of economic and social systems (Konsbruck, n.d.). We cannot live without mobile phone technology, credit data systems, electronic money transfer systems, etc. If the world's computer systems failed then so would our food and power distribution networks.

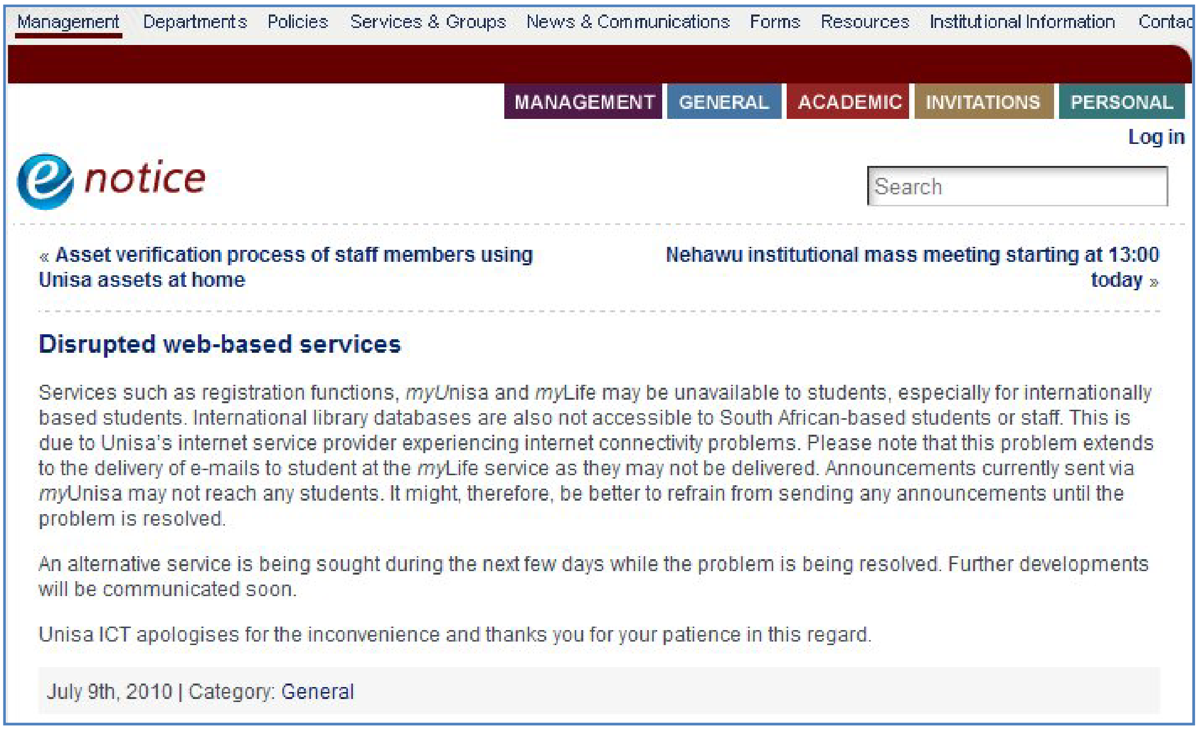


Figure 5.2 A message to Unisa staff about unavailability of web-based services

Figure 5.2 shows a notice that went out to Unisa staff about the unavailability of web-based services, including myUnisa. A relatively small problem such as this can have far-reaching consequences. Students who rely on the electronic submission of assignments may miss due dates of assignments. If such a problem persists, the university may be compelled to give a general extension on assignment due dates, interfering with the tuition schedules of individual modules. Assignments may not be marked and returned to students in time before the examination starts.

5.3 Social Networking Technologies

In Unit 1 (section 1.2.5.4) we briefly introduced social network web sites (also referred to as online communities). In this section we take that discussion further, looking also at other types of social networking tools, such as blogs and instant messaging systems.

5.3.1 Chat Rooms

Chat rooms are locations on the Internet where people meet to have online conversations in real time. They are usually open for anyone to join and often relate to a specific topic (for example, a support group for people with a specific medical condition). The messages are normally visible to all current visitors but users can participate in private conversations not observable by others. Some chat rooms allow users to publish their photographs and personal information, or use Web cameras. There may be rules for participation and the content may be monitored, but many have no restrictions on what may be discussed or in what kind of language. There is a danger that young users can be targeted by sexual predators. Ybarra and Mitchell (2007) report that chat rooms have been losing popularity among teenagers since 2000 with many of them regarding these rooms as unpleasant places.

5.3.2 Instant messaging (IM)

The South African MXit system is an example of an IM system. It is a real-time communication tool that allows two or more users who are connected to the system to interact with each other synchronously. IMs are different from chat rooms in the sense that the sender must know the user name of the recipient to send a message. Some IM services have a searchable member directory where users can add information so that other users of the system can identify them to send a message (Ybarra and Mitchell, 2007). Privacy settings make it possible to block out messages from unknown users or messages from specific individuals.

5.3.3 Blogs

Blogs are like online journals. Individuals use them as diaries or to comment on specific topics. Some allow readers to post responses. Blogs are also popular amongst children. In 2009, 24% of all children between 9 and 16 in the United Kingdom had their own blog (New Scientist, 12 December 2009).

5.3.4 Social Networking Sites

Social networking sites are sometimes referred to as web communities or online communities. These sites integrate all of the tools above. Users create profiles with descriptive personal information and photographs and write messages on message boards or walls (as in a blog). Profiles are interconnected through explicitly declared friend relationships (Caverlee and Webb, 2008). Users communicate through different kinds of messaging mechanisms. Communication can be synchronous (like 'chatting') or asynchronous (like blogs or e-mail). Users can limit access to their profiles through privacy settings or they can leave it public and allow anybody to view their information.

It has been possible to publish personal information on the WWW for a long time. The uncomplicated mechanisms that social networking sites provide to do this have now caused an explosion of detailed personal data on the Internet. The success of social networking sites can be attributed to the fact that they provide a standardised, centralised, easy and free way to create an Internet presence (Jones and Soltren, 2005).

A social network in the general sense refers to all the people one has a social relationship with. Online social networks, however, include 'friends' whom the user have never met and have no other link to besides the fact that they appear as 'friends' on their profile. Users of social network sites typically communicate directly with a very few individuals in their friend lists (Huberman et al., 2008).

An important issue linked to online social networks is privacy. It seems that for some time after Facebook emerged, users were ignorant about the consequences of excessive disclosure of personal information (Jones and Soltren, 2005). Caverlee and Webb (2008) report that there is a steady growth in the use of privacy settings by new members on MySpace – one of the most popular online communities. This indicates that people are becoming more aware of the privacy risks associated with these sites.

Online social networking has influenced the way people interact and relate to one another. It is a cheap and easy way for family and friends to stay in touch while physically removed. Introverts who, in the physical world, would seldom meet new people and make friends can now build online relationships from within the 'safe' environment of a web-based community. It can, unfortunately, become an addiction that compels people to constantly check for Facebook updates or Twitter messages. I recently spent a weekend in Zambia on the bank of the breath-taking Kariba lake with some friends. In our party were three women in their late twenties and early thirties who were inseparable from Facebook. For me the weekend inadvertently became an HCI research project – an observation of the three women's interaction with technology. They were completely incapable of appreciating the beauty of the lake and its surroundings by just looking at it. Every view had to be photographed and immediately loaded into a Facebook photo album. They then gathered around a laptop and enjoyed their holiday by browsing the photographs of it. It was obvious that they would have been at a complete loss if their cell phones and notebook computer were taken away.

Some advantages of social networking sites:

1. Low cost of creating a web presence.
2. Making personal connections – for example, by searching for people who share your interests or becoming friends with 'friends of friends'. You can also reconnect with long lost friends. For many people social network sites are their primary mechanism to find a date.
3. Making connections for career purposes – it is quite easy to identify people who work in your field by searching through their profiles.
4. For businesses it has the advantage that they can get additional information on someone before

employing them. It is a way to find out if people have lied in their applications or CVs.

Some disadvantages of social networking sites:

1. Lack of anonymity or privacy.
2. Identity theft – some people place enough information on these sites to allow others to get all the necessary information to assume that identity.
3. It wastes time, to such an extent that some companies block access to these sites during working hours.
4. Mining of users' data for advertising purposes.
5. Cyberbullying – it is much easier to harass someone through an online network than it is in the real world.
6. Cyberstalking.
7. Inappropriate content such as political propaganda. Countries such as Syria, China, Iran, and Vietnam have banned the use of Facebook.

ACTIVITY 5.6
List five more advantages and five disadvantages of social networking sites. Use examples from your own experience using these sites.

5.4 The Digital Divide

The digital divide refers to unequal access to technology that separates people into those who have it and those who do not (Attewell et al., 2003). Some of the contributing factors are financial constraints, lack of skills, unavailability of basic infrastructure (e.g. electricity) and carelessly designed systems. In developing countries where technology and internet access are relatively widely available, there are still problems with fast internet access. Specific problems are associated with online interaction, especially where bandwidth is limited. Internet content providers often do not show consideration for these limitations and will not compromise, for example, on the use of sound and graphics that can take a long time to download. This means that Internet-based applications are unusable by a large part of the potential user population. Literacy levels also contribute to the digital divide.

Lack of adequate cognitive resources is an important contributor to the digital divide (Wilson, 2006). Interacting with computers requires basic skills that will enable a user to recognize the need for information, find the information, process and evaluate the information for its appropriateness, and apply it in a meaningful way.

Figure 5.3 Children using a Digital Doorway workstation

The digital divide is not only a reflection of the separation between developed and developing economies. It can also exist among population groups within the same nation. In the United States, white and Asian people are at least 20% more likely to own computers than black and Hispanic people (Cooper and Kugler, 2009). In 2001 only 2% of black households in South Africa had computers compared to 46% of white households (Statistics South Africa, 2001).

ACTIVITY 5.7

UniNet is a South African telecommunications company. They have made a documentary on the project entitled UniFi Knysna, through which the whole of Knysna was supplied with Wi-Fi (Wireless Internet). Watch the video at the following link: http://www.youtube.com/watch?v=TpxCm8Snt9I.

Many attempts are being made to bridge the divide, for example the Digital Doorway project (Meraka Institute, Accessed 23 Oct 2007) in South Africa, MIT's One Laptop Per Child project (MIT, Accessed 23 Oct 2007) and the Hole-in-the-Wall project in India (Mitra, 2003). The Digital Doorway project is a joint initiative by the Department of Science and Technology and Meraka Institute of the CSIR and focuses on providing physical computers to underprivileged communities in South Africa. Digital Doorways are non-standard computer systems housed in rugged, custom-designed kiosks with multiple terminals that can be accessed simultaneously by users (see Figure 5.3). The robust housing and metal keyboard protects the system against vandalism. The aim is to promote computer literacy through unassisted learning (Cambridge, 2008, Gush et al., In Press) by installing the computers in underprivileged communities such as schools, police stations and community centres around South Africa. By mid-2010, 206 Digital Doorways had been deployed around the country.

ACTIVITY 5.8

Do research on the Internet about the One Laptop Per Child (OLPC) project. Write a one-page essay on this project that includes at least the following information:

- Who initiated it and when?
- What does the project involve?
- Where has it been deployed?
- How successful is it?
- What problems are associated with the project?

5.5 Conclusion

Many books have been written on social issues in human-computer interaction and on the impact of technology on society, so it is difficult to reduce these topics to a discussion in one unit of a study guide. Our aim was to give you an idea of the profound effects of technological advancement on society. We only touched on aspects such as privacy and security and did not even mention important issues such as ethics and intellectual property. We do hope that we have stirred you interest and that you will, as future designers or IT managers, keep yourself informed of the pervasive impact that computers have on our world.

REFERENCES

ACM SIGCHI (n.d.) *Curricula for Human-Computer Interaction*, http://www.sigchi.org/, accessed 24 June 2010.

ANTLE, A. (2007) Designing Tangibles for Children: What Designers Need to Know. *CHI 2007.* San Jose, California, USA, ACM Press.

ATTEWELL, P., SUAZO-GARCIA, B. & BATTLE, J. (2003) Computers and Young Children: Social Benefit or Social Problem? *Social Forces,* 82**,** 277-296.

BAECKER, R. M. & BUXTON, W. A. S. (Eds.) (1987) *Readings in Human-Computer Interaction: A Multi-Disciplinary Approach*, Los Altos CA, Morgan Kaufman.

BARTLETT, B., ESTIVILL-CASTRO, V. & SEYMON, S. (2004) Dogs or Robots: Why do Children see them as Robotic Pets rather than Canine Machines? IN COCKBURN, A. (Ed.) *Conference on Research and Practice in Information Technology.*

CAMBRIDGE, G. (2008) Digital Doorway: Enriching your Mind. *2nd CSIR Biennial Conference.*

CARROLL, J. M. (2003) Introduction: Toward a Multidisciplinary Science of Human-Computer Interaction. IN CARROL, J. M. (Ed.) *HCI Models, Theories and Frameworks.* San Fransisco, Morgan Kaufman Publishers.

CARROLL, J. M. (2009) *Encyclopedia entry on Human Computer Interaction (HCI),* http://www.interaction-design.org/encyclopedia/human_computer_interaction_hci.html, accessed 22 June 2010.

CAVERLEE, J. & WEBB, S. (2008) *A Large-Scale Study of MySpace: Observations and Implications for Online Social Networks*, accessed 1 July 2010.

COOPER, J. & KUGLER, M. B. (2009) The Digital Divide: The Role of Gender in Human-Computer Interaction. IN SEARS, A. & JACKO, J. A. (Eds.) *Human-Computer Interaction: Designing for Diverse Users and Domains.* Florida, Taylor & Francis Group, LLC.

DARZENTAS, J. & MIESENBERGER, K. (2005) Design for All in Information Technology: A universal Concern. IN ANDERSON, K. V., DEBENHAM, J. & WAGNER, R. (Eds.) *Lecture notes in Computer Science.* Springer-Verlag.

DIX, A., FINLAY, J., ABOWD, G. & BEAL, R. (2004) *Human-Computer Interaction*, Pearson Education Ltd.

DONG, Y. & LEE, K. (2008) A Cross-Cultural Comparative Study of Users' Perceptions of a Webpage: With a Focus on the Cognitive Styles of Chinese, Koreans and Americans. *International Journal of Design,* 2.

DRUIN, A. (1996) CHIKids: A Common Ground for Kids and Adults. IN TAUBER, M. J. (Ed.) *Common Ground: CHI 96 Conference Proceedings.* ACM.

GAME VORTEX (2008) *EyeToy Play 2*, www.psillustrated.com/gamevortex/soft_rev.php/2686, accessed 22 January 2008.

GATHERCOLE, S. E. (2002) Memory Development During the Childhood Years. IN BADDALEY, A. D., KOPELMAN, M. D. & WILSON, B. A. (Eds.) *The Handbook of Memory Disorders.* Second ed., John Wiley & Sons, Ltd.

GUSH, K., DE VILLIERS, M. R., SMITH, R. & CAMBRIDGE, G. (In Press) Digital Doorways. IN STEYN, J., BELLE, J. & VILLENEUVA, E. M. (Eds.) *Development Informatics and Regional Information Technologies: Theory, Practice and the Digital Divide.* Pennsylvania, IGI.

HARPER, R., RODDEN, T., ROGERS, Y. & SELLEN, A. (Eds.) (2008) *Being Human: Human-Computer Interaction in the Year 2020,* Microsoft Research Ltd.

HENRY, S. L. (2002) Understanding Web Accessibility. IN THATCHER, J., BOHMAN, P., BURKS, M. R., HENRY, S. L., REGAN, B., SWIERENGA, S., URBAN, M. D. & WADDELL, C. D. (Eds.) *Constructing Accessibility Web Sites.* Birmingham, Glasshaus.

HENRY, S. L. (2007) Just Ask: Integrating Accessibility Throughout Design. Madison, WI, ET\Lawton.

HORNECKER, E. & BUUR, J. (2006) Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction. *CHI 2006.* Montreal, Quebec, Canada.

HUBERMAN, B. A., ROMERO, D. M. & WU, F. (2008) *Social networks that matter: Twitter under the*

microscope,arXiv:0812.1045v1 [cs.CY] 4 Dec 2008.
HUGHES, J. (2001) *Sony AIBO,* http://the-gadgeteer.com/review/sony_aibo_review, accessed 22 January 2008.
HUTCHINSON, H., DRUIN, A. & BEDERSON, B. B. (2007) Designing Searching and Browsing Software for Elementary-Age Children. IN LAZAR, J. (Ed.) *Universal Usability: Designing Computer Interfaces for Diverse Users.* West Sussex, John Wiley and Sons Ltd.
INKPEN, K. M. (1997) Three Important Research Agendas for Educational Multimedia: Learning, Children and Gender. *AACE World Conference on Educational Multimedia and Hypermedia 97.* Calgary.
ISYS (2000) *Making Information Usable: Interface Hall of Shame,* www.iarchitect.com/mshame.htm, accessed.
JOHNSON, C. (1997) *Interactive Systems Design,* http://www.dcs.gla.ac.uk/~johnson/teaching/isd/course.html, accessed 24 June 2010.
JONES, H. & SOLTREN, J. H. (2005) *Facebook: Threats to Privacy,* http://groups.csail.mit.edu/mac/classes/6.805/student-papers/fall05-papers/facebook.pdf, accessed 20 June 2010.
KAEMBA, M. (2008) *Digital museum usability: A study with elderly users,* ftp://cs.joensuu.fi/pub/Theses/2008_MSc_Kaemba_Mitwa.pdf, accessed 18 April 2009.
KONSBRUCK, R. L. (n.d.) *Impacts of Information Technology on Society in the new Century,* http://www.zurich.ibm.com/pdf/Konsbruck.pdf, accessed 1 July 2010.
KRAUS, L., STODDARD, S. & GILMARTIN, D. (2006) *Chartbook on Disability in the United States,* http://www.infouse.com/disabilitydata/disability/1_1.php, accessed 4 August 2009.
LEHOHLA, P. (2005) *Census 2001: Prevalence of Disability in South Africa,* http://www.statssa.gov.za/census01/html/Disability.pdf, accessed 22 July 2009.
LEPAGE, M., QIU, X. & SIFTON, V. (2006) *A New Role for Weather Simulations to Assist Wind Engineering Projects,* http://www.rwdi.com/cms/publications/47/t28.pdf, accessed 9 July 2010.
MERAKA INSTITUTE (Accessed 23 Oct 2007) *The Digital Doorway Project,* www.meraka.org.za/digitalDoorway.htm, accessed.
MIT (Accessed 23 Oct 2007) *One Laptop per Child Project,* www.laptopical.com/2b1-laptop.html, accessed.
MITRA, S. (2003) Minimally Invasive Education: A Progress Report on the "Hole-in-the-Wall" Experiments. *British Journal of Educational Technology,* 34**,** 367-371.
MONTEMAYOR, J., DRUIN, A. & HENDLER, J. (2000) PETS: A Personal Electronic Teller of Stories. IN DRUIN, A. & HENDLER, J. (Eds.) *Robots for Kids: Exploring New Technologies for Learning.* Morgan Kaufman.
MUGLIA, B. (2010) *Modeling the World,* http://www.microsoft.com/mscorp/execmail/2010/05-17HPC.mspx, accessed 9 July 2010.
MYERS, B. A. (1998) A Brief History of Human Computer Interaction Technology. *ACM Interactions,* 5**,** 44-54.
NIELSEN, J. (1994) Heuristic Evaluation. IN NIELSEN, J. & MACK, R. (Eds.) *Usability Inspection Methods.* New York, John Wiley and Sons.
NIELSEN, J. (2001) *Ten Usability Heuristics,* www.useit.com/papers/heuristic, accessed.
NISBETT, R. E. (2003) *The geography of thought: How Asians and Westerners think differently - and why.,* New York, Free Press.
NORMAN, D. A. (1999) *The Design of Everyday Things,* MIT Press.
Deign, J., ROGERS, Y. & SHARP, H. (2007) *Interaction Design: Beyond Human-Computer Interaction,* John Wiley & Sons, Inc.
REASON, J. (1990) *Human error,* Cambridge University Press.
SHNEIDERMAN, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* New York, Addison-Wesley.
SIMONITE, T. (2010) Online Translation Services Learn to Bridge the Lnaguage Gap. *New Scientist.*
STATISTICS SOUTH AFRICA (2001) *Census 2001: Key Results,* http://www.statssa.gov.za/census01/HTML/Key%20results_files/Key%20results.pdf, accessed 03 May 2010.
SWAMINATHAN, N. (2007) Could Robots Become Your Toddler's New Best Friend? *Scientific American.*
UNITED NATIONS (2006) *World Programme of Action Concerning Disabled Persons,* http://www.un.org/esa/socdev/enable/diswpa01.htm, accessed 30 September 2009.

UNIVERSITY OF MARYLAND HCI LAB (2008) *Project Screenshots*, www.cs.umd.edu/hcil/pubs/screenshots/PETS, accessed 22 January 2008.

WADDELL, C. D. (2002) Overview of Law and Guidelines. IN THATCHER, J., BOHMAN, P., BURKS, M. R., HENRY, S. L., REGAN, B., SWIERENGA, S., URBAN, M. D. & WADDELL, C. D. (Eds.) *Constructing Accessible Web Sites.* Birmingham, Glasshaus.

WILLIGES, B. H. & WILLIGES, R. C. (1984) User Considerations in Computer-Based Information Systems. IN MUCKLER, F. A. (Ed.) *Human Factors Review 1984.* Santa Monica, Human Factors Society.

WILSON, E. J. (2006) *The Information Revolution and Developing Countries*, Cambridge, MIT Press.

YBARRA, M. L. & MITCHELL, K. J. (2007) *How Risky Are Social Networking Sites? A Comparison of Places Online Where Youth Sexual Solicitation and Harassment Occurs*, www.pediatrics.org/cgi/doi/10.1542/peds.2007-0693, accessed 1 July 2010.