Textual Cognetics and the Role of Iconic Linkage in Software User Guides

Jody Byrne BA (Hons)

A thesis submitted to Dublin City University for the degree of Ph.D. in Applied Languages

Supervisor: **Dr. Dorothy Kenny**

Centre for Translation & Textual Studies,
School of Applied Language & Intercultural Studies,
Dublin City University

July 2004

DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of *Ph.D. in Applied Languages*, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:	Jody Byrne	Jody Byrne (Candidate)
ID No.:	92243851	
Date:	5th July, 2004	

ACKNOWLEDGEMENTS

Get the advice of everybody whose advice is worth having – they are very few – and then do what you think best yourself

Charles Stewart Parnell

The decision to embark upon Ph.D. research was one which brought with it much excitement, trepidation and occasionally a little self-doubt. This project has been a tremendous learning experience, not only in academic terms but also in the sense that I discovered some remarkable qualities in those people around me who have helped and supported me through this challenging period. I would like to thank them here.

My utmost gratitude goes first and foremost to my supervisor Dr. Dorothy Kenny. Without her guidance, advice, patience, support and much needed pep talks when panic and self-doubt set in, this Ph.D. would have been a considerably more arduous and traumatic pursuit. While Dorothy did everything she could to keep my research focussed and on-track, ultimate responsibility for any shortcomings in this thesis rests entirely with me.

I would also like to thank my colleagues at the *Centre for Translation & Textual Studies* for their friendship, support and help as well as for providing facilities for the usability study. My thanks also go to the *School of Applied Language & Intercultural Studies* at Dublin City University for funding my research despite severe budget restrictions.

When it seemed that I spent every waking moment buried in books or in front of a computer, and when my spirits were at low ebb, my family were always there to give me support and advice and to cheer me up and drag me away from the computer so that I had a life outside my research. Despite the trials and tribulations of the past number of years, Mum, Gran, Sinéad and Willie have managed to keep me motivated and relatively sane for the duration. You really are the best family in the world! My thanks also go to the late Joe Howlett, my favourite uncle who missed the completion of this Ph.D. by a few short months.

I would especially like to thank Jan, the most precious part of my life. Her love, support and encouragement as well as her incredible ability to put up with frequent neglect and the worst of my mood swings (and occasional tantrums) have made it all worthwhile. Without Jan, there would have been a lot more tears and a lot less laughter over the past few years. Thanks for putting up with me!

My thanks also go to the following:

- *Digitake Software Systems Ltd.* for their incredible generosity in allowing me to use their software and documentation and for making the usability test possible
- *The supervisors and administrative staff of the Fianna Fail parliamentary party* at Leinster House for generously agreeing to participate in the main usability study
- The students of the MA in Translation Studies programme (2002/2003) for participating in my pilot study
- *Fiona O'Callaghan* for her invaluable assistance in conducting the statistical analyses of the results of my usability study
- *Francoise Blin* in the School of Applied Language & Intercultural Studies at Dublin City University for her support and for introducing me to Camtasia
- *Dr. Annette Simon* and *Dr. Jenny Bruen* for their valuable advice
- *Robert Duffy* from DCU Educational Services for his much appreciated assistance with the video camera

CONTENTS

LIST OF FIGURES

List of Tables

ABSTRACT

This study investigates whether Iconic Linkage - the use of the identical wording to present the same information recurring in a text - can improve the usability of user guides. Drawing on research literature in technical communication, cognitive psychology and human-computer interfaces, Iconic Linkage is presented as a writing strategy that potentially allows users to work more quickly and effectively and which promotes better retention of information. The usefulness of Iconic Linkage was tested in a laboratory-based usability study that combined (1) objective task-based evaluation and (2) users' subjective evaluations of a software program used in recording parliamentary debates. A post-test survey designed to test subjects' retention of information contained in the user guides was also administered. The study shows that Iconic Linkage significantly improved usability of the user guide: in all tasks, subjects worked more effectively and made fewer mistakes; while in the three timed tasks, subjects completed the tasks much more quickly. Subjects also gave higher ratings for the software and their retention of information was noticeably improved. The study concludes by discussing the implications and potential future applications of this research.

1.1 INTRODUCTION & CONTEXT

Mobile phones, video games, digital cameras, MP3 players, word processing software, televisions, x-ray machines, satellite navigation systems, DVD players and industrial process control systems. A reliance on semi-conductors notwithstanding, a common theme linking this diverse range of products is that they are all accompanied by some sort of user guide. More specifically, they invariably include a software user guide.

A common misconception about software user guides is that they are written only by software companies for software products. In reality, however, any company that produces software - even if it is only as a supplement to the company's main product - will produce software user guides (Van Laan & Julian 2001:4).

Despite the wide-scale production of user guides, especially in the software and technology industries, it appears quantity has not translated into quality. Indeed, poor or inadequate user guides are a widely acknowledged problem in industry (Brockmann 1990:1). That there exists such inadequate documentation is disturbing, especially when we consider that aside from certain legal implications, the quality of user guides can spell success or failure for a product or even for a company. One such documented example refers to the catastrophic losses incurred in 1983 by a company called Coleco. This company lost a staggering US$45 million in the final three months of 1983 as thousands of irate customers returned the Coleco Adam computer, citing the terrible user guide as the problem (Brockmann 1990:13). Stories like this are numerous and it is clear that user guides can help improve sales and create loyal customers (as was the case with Apple computers in the 1970s and 1980s).

But high quality user guides are not just useful ways of gaining new customers. As products become more and more sophisticated and complex, it is essential that quality user documentation is available to help users exploit the full range of functions offered by the product. Companies frequently spend vast sums of money on products of which only a fraction of the functions will ever be used.

It is possible that the problems of poor documentation are due to the simple fact that companies do not always understand the purpose of user guides and those who produce them, not to mention the needs of the customers who use the user guides.

User guides are, in effect, an interface between computer systems and their human users. In producing user guides, technical communicators need to act as an intermediary between the software and the users. To be successful in producing user guides, it is essential not only to understand the product in detail, but also to understand the humans who will use it. Coe (1996:2) states that technical communicators design information for users and there is "a covenant of trust" between communicator and user. This, she maintains, is the basis for human factors in user documentation.

Indeed, in recent years, the lines between documentation and software have become somewhat blurred with the advent of single-source, multi-channel publishing whereby a single stock of text is produced for use in a variety of media such as printed documentation or online help etc. In addition, there has been a trend which has seen manufacturers provide documentation in electronic form only, for example, in the form of PDF files. Some software, does not come with what is traditionally regarded as documentation, favouring instead, comprehensive help systems with complex interfaces which allow task-specific items of information to be accessed directly from within the software.

This has been justified for a number of reasons, most notably those relating to cost. There are many arguments that producing print documentation is considerably more expensive than producing online documentation. However, on closer examination, such arguments are less than convincing. Starr (2001) and Curwen (2002) maintain that the question of costs is less to do with actual costs and more to do with ***who*** bears the costs of printed documentation. They argue that while manufacturers generally escape relatively unscathed by distributing documentation on a "CD-ROM that you know cost them around one dollar to manufacture" (Starr

2001), users ultimately end up printing the documentation on an inkjet or laser printer; the cost to the user can frequently be double the cost the manufacturer would have incurred.

But the real reason why justifications for the proliferation of online documentation are inadequate lies in the reason why users feel the need to print off sometimes entire chapters of the online documentation: they frequently find it too hard to read from a computer screen. The fact that it is commonly believed that users take 20-30% longer to read from a screen than from a page (Curwen 2002) is based on established research by the likes of Dillon (1992). So while the provision of printed documentation is ostensibly assuming less importance for software manufacturers, for users, they continue to be essential.

Admittedly, in recent years, the quality of user guides has improved steadily. Yet there are huge numbers of people who simply do not read user guides no matter how complex the products they want to use. In fact, it seems that sometimes the more complex the product, the less likely some people will be to read the user guide. While we can attribute this to an expectation by users that modern products are generally intuitive and self-explanatory, it is more likely that we are dealing with people who Coe claims have "lost their trust of technical communications" (1996:3). For them, "past experiences may have destroyed their trust and colored their approach to and use of" technical information presented in, for example, user guides (*ibid.*).

Here, the problem facing user guides is more serious and more difficult than simply teaching users how to use a product. Rather, the task is to re-establish contact and trust with users and persuade them to read and trust user guides. These users frequently have just reason to be wary of user guides because previous experiences have left them feeling frustrated, confused or just plain stupid. An interesting discussion of this issue is provided by Schriver (1997:214-222) who cites feelings of confusion and incompetence among users as a result of inadequate instructions. Interestingly, users generally blame themselves for their inability to follow instructions for products (ibid:222).

The Cost of Inadequate User Guides

The consequences of inadequate user guides should not be underestimated. Approximately half of all product returns and complaints in Germany arise as a result of bad instructions. Frequently, customers end up damaging products themselves because they lose patience with bad user guides and resort to "creative" and sometimes unorthodox methods to get their products to work (Cognitas 2003a). The resulting damage and compensation amounts to some €500 million each year in Germany alone (*ibid.*). This is due in large part to changes in European Union laws governing product liability. European Resolution *C411* states that

> ...inadequate operating instructions may affect the presentation of products and may be a factor to be taken into account together with all other pertinent circumstances in considering whether goods are defective (Council of the European Union 1998:1)

This resolution goes on to say that in the light of the wider range of products and the advances being made in technology...

> ...operating instructions for technical consumer goods are often perceived by consumers as inadequate, both because they are unclear and present language difficulties, owing to faulty translations or to the use of terms which are too complex, and because they lack structure and have inadequate content.

Tackling the Problem of Poor User Guides

The requirements in this resolution have filtered down into national laws. For example, in early 2002, Germany's product liability law (*Produkthaftungsgesetz*) was overhauled with the result that user guides are regarded as a part of the product and as such, any defects or faults they contain are regarded as product defects which can result in the rejection or withdrawal of the product (Heino 1992:111). In addition, where a product is damaged or destroyed as a result of a user following instructions contained in an inadequate user guide, the manufacturer or retailer is obliged to provide a replacement (Cognitas 2003b).

To counteract the problem of poor user guides, the European Union has codified what it believes to be the essential characteristics of "good" user guides. Resolution C411 (Council of the European Union 1998) sets out, among other things, a series of criteria under the following headings, which the Council of Europe believes will make for more effective user documentation:

- *Development of documentation*: all relevant laws, standards and guidelines should be consulted and the document must comply with their requirements
- *Content of documents*: the content of documents should be structured logically and reflect real use; warnings and cautionary information must be clearly distinguishable from the main document content
- *Separate documents for different models*: unless the procedures involved in using functions are identical, separate documents must be produced for different models or variations of products
- *Safety and warning instructions*: must be clear and easily accessible
- *Document language*: user documentation must be available in a user's own language
- *Style and layout*: should ensure clear and readable documents

The overall aim of this is to produce high quality documentation which will provide customers with "adequate user information to ensure proper and complete use of the product" (Council of the European Union 1998:1). Other regulatory and legislative tools governing the production and provision of user guides include:

- *EN 62079*. "Preparation of instructions - Structuring, content and presentation"
- *EN 292-2* "Safety of machinery. Basic concepts, general principles for design"
- *VDI 4500-2* "Technical documentation - Internal technical product documentation"

These standards and guidelines have gone some way towards ensuring better user guides are provided to users. One initiative, based in part on these regulations, is the *DOCcert* certification scheme developed in 1993 by *tekom* and *TÜV* in Germany (Jung & Becker 2003). This is a quality assurance and certification programme aimed at ensuring documentation is effective, complete and facilitates the safe use of products. The certification process tests documentation for comprehensibility, completeness and safety and takes place in three stages.

The first stage involves examining relevant laws, standards and guidelines such as those mentioned above and ensuring that the documentation complies with their requirements. The second stage involves testing the documentation on the basis of a series of criteria such as accuracy, comprehensibility, layout, readability etc. The final stage involves hands-on usability testing with users. Successful documentation is then certified by *TÜV* and can bear the *TÜV*-approved logo shown in Figure 1.

Figure 1: DOCcert label for certified user documentation (www.cognitas.de)

These initiatives notwithstanding, it is clear that work on improving the quality of user guides is far from complete and that there are still countless inadequate user guides in circulation. A study conducted by the German computer magazine *ComputerBild* in 1999 examined 60 user guides from a range of well-known companies and found that 35 could be regarded as "faulty" and could result in complaints or claims for compensation (ComputerBild 1999:16). Using the *DOCcert* test procedures and criteria, the investigators found that only 4 of the user guides passed the stringent requirements.

The obvious need to overhaul the way in which user guides are produced has serious implications for vast numbers of technical communicators across the world. Up until now, we have referred to technical communicators as being responsible for the production of user guides. While traditionally *technical communication* would be taken to mean technical writers alone, the industry and nature of the work have developed to a point where technical communication includes the work of technical writers, illustrators, technical translators, editors and web designers (Van Laan & Julian 2001:5). Indeed, many professional technical communication associations explicitly include these roles under the umbrella term of *technical communication*. Given the fact that according to Council of the European Union *Resolution C411* "customers are entitled to manuals produced in their own language" (Council of the European Union 1998:3), it is clear that "translation work [is] an integral part of the process of creating technical documentation" (Budin 2000).

Technical writing and technical translation are inextricably linked with regard to user guides. As such, any discussion of user guide quality must take this relationship into account, not least because translation is explicitly identified in the aforementioned European Directive as a potential cause of problems in user guides (Council of the European Union 1998: 1).

Historical Problems in Document Usability Research

The preceding paragraphs make it clear that the quality of user guides is of paramount importance; in this study we will examine document quality from a usability point of view. A review of literature on usability evaluation reveals a range of publications concerned with improving the usability of software documentation. However, much of the research undertaken in recent years is less than satisfactory for a number of reasons.

First of all, the term *documentation* as used by several authors proves problematic, not only in terms of the aims of this study, but also in terms of what happens in an industrial context. Documentation is frequently defined in an extremely broad sense as including practically anything that involves some form of text. Indeed, Mamone (2000:26) defines documentation as

> ...user manuals, online help, design features and specifications, source code comments, test plans and test reports, and anything else written that explains how something should work or be used.

While this is perfectly acceptable and indeed suitably comprehensive in many ways, it is problematic from the point of view of usability evaluation. Mamone's assertion (*ibid.*) that documentation can come in "hard or soft form" fails to take into account the fact that while technology and economics have pushed the online delivery of information, online and hardcopy documentation have specific strengths and weaknesses (Smart & Whiting 1994:7). As such, they cannot be assessed using identical sets of criteria.

Indeed, the all-inclusive definition of documentation includes online help. In addition to the obvious textual issues involved in using help and the fact that information is presented as short, independent units, the diversity and sophistication of help systems mean that there is quite a different set of considerations (such as navigation models, resolution, display speeds, download times, delivery options etc.) to be borne in mind when evaluating their usability in comparison with that of hardcopy documentation. So while Mamone provides a useful overview of usability test procedures, the failure to identify the differences and the similarities between hardcopy documentation and online help compromises his approach. But it is not just Mamone who fails to make this distinction, Prescott & Crichton (1999), Harrison & Mancey (1998), Simpson (1990) and Mehlenbacher (1993), to name a few, either concentrate on online texts or group online texts together with print documentation.

Although the practical issues relating to the evaluation of online documentation mean that online and print documentation cannot reasonably be assessed using a single theoretical framework, other factors justify the separate treatment of the two types of text. By grouping print and online texts together, print documentation risks being regarded as a "low-tech" form of documentation which can be assessed with just a subset of the online evaluation paradigm. Admittedly, online documentation is increasingly being regarded as an integral part of products which speeds up the dissemination of information and which – if designed well – can allow users to access information more quickly. Nevertheless, a large proportion of users prefer the "book" format. According to Smart & Whiting(1994:7) "some information is best accessed from a printed form." Such information includes trouble shooting, lengthy conceptual descriptions or material where annotation is essential. With this in mind, we can see the need to examine documentation purely from a print point of view.

Another problem relating to previous research into usability evaluation, and one which explains the trends outlined in the paragraphs above, is the fact that usability evaluation of both online and print documentation appears to have become unfashionable. During the 1980s, according to Weiss (1995:3), software developers and technical writers shared a common goal: make products and their manuals as usable as possible, i.e. as easy to learn and operate as possible. The approach adopted what Weiss terms "paternalistic" tendencies in that developers and documentors pre-empted problems and restricted the possible choices (and thus incorrect choices) open to users so as to minimise the likelihood of users making mistakes. This approach contrasted sharply with the attitudes of the 1960s and 1970s when software was regarded as inherently and incurably difficult. In the 1980s, however, this view gave way to a more caring approach and the enormous and voluminous reference manuals to be used by users of **all** levels were superseded by "friendly" user manuals which provided clear, easy-to-follow, task-orientated instructions.

Over time, the aim of developers became the production of intuitive interfaces. By "replacing remembering with recognizing [and] by replacing the typing of long strings with short strings… and reducing the causes of confusion and impasse" (Weiss 1995:4) it was believed possible to produce computers and software whose users rarely, if ever, needed to use a manual. Eventually, it became widely accepted that products that required users to consult manuals were "bad" products which were badly designed (ibid.).

This perception continued and, in the 1990s, the advent of GUI-based computing revolutionised the face of computers and software (Weiss 1995:8). These systems were heralded as the ultimate in usability, and in comparison to the intimidating command line systems previously available, this was probably true at the time. However, the emphasis of developers shifted away from usability – after all, it appeared that this particular dragon had been slain and usability was no longer an issue. Consequently, because it was believed that the products and systems were extremely usable and intuitive, the importance of and need for manuals diminished significantly. Another reason may be the perception that online documentation is cheaper to produce and disseminate. Starr (2001) presents some interesting ideas to counter this notion, maintaining, among other things, that the costs are, in fact the same and that the difference is who ultimately pays for the materials.

With the emphasis of software manufacturers firmly taken away from usability and placed on the new found power, flexibility and advanced features presented by graphical interfaces and higher powered computers, the priorities of technical communicators also changed as they now had to contend with increasingly sophisticated information delivery methods **and** products that offered a bewildering array of features and customising options (Weiss 1995:9). Somewhere along the line, many people lost sight of the fact that the more complex products become, the greater the need to ensure usability. As Weiss (*ibid.*) says, "these baroque indulgences are inconsistent with the classic notion of user-friendliness." And while the abundance of features and customisability have "all but murdered" the notion of inherent usability the belief that software should be instantly learnable without manuals is stronger than ever (Weiss 1995:13; Mehlenbacher 1993:210).

This situation, combined with new delivery technologies has forced technical communicators to justify their existence and importance in the production of software. Consequently, much of the research carried out in the last two decades has centred on the content of documentation and online methods for its delivery. Those studies that have been carried out on usability have either been defined by the new and somewhat misguided reprioritisation of flexibility over real usability or they have been preoccupied with the challenges of online delivery methods. Others, such as Mehlenbacher (1993), Spool (1996), Prescott & Crichton (1999) and Novick (2000) examine usability from a developmental viewpoint rather than from a purely diagnostic point of view.

Moreover, in an evaluation of 22 usability studies, Mirel (1990) points out that none have been specifically designed to provide standardised, quantifiable evaluations of documentation, much less for software user guides. Mirel also maintains that many of the studies adopted incoherent approaches to evaluations. Unfortunately, the approach put forward by Mirel is almost forensic in nature and is of little use here because we simply want to determine whether something affects usability and if so, how it does this. Nevertheless, Mirel does attempt to impose some form of order and standardisation in what is a largely chaotic, ad-hoc and uncoordinated area of research.

Notwithstanding the difficulties outlined in the previous paragraphs, a number of studies exist which provide varying levels of insight into the practicalities of conducting usability evaluations. These studies will be discussed here in due course.

1.2 Aims & Structure of this Study

The aim of this study is to investigate methods for improving the quality of software user guides. More specifically, this study seeks to establish whether writing identical information which recurs throughout a text using precisely the same wording can make user guides more usable. This practice of phrasing semantically identical information in the same way throughout a text is referred to in this study as *Iconic Linkage*.

Iconic Linkage is a textual strategy that can be implemented during both the composition and editing stages of user guide production. And because it is a strategy for formulating sentences, it is also applicable to translation. Consequently, this study seeks to establish whether the presence of Iconic Linkage makes user guides – whether original monolingual versions or translations – more usable and effective.

As mentioned above, the close proximity of technical writing and technical translation in industry means that our discussion here will need to take into account the fact that user guides are the product of **both** technical writers **and** technical translators. Nevertheless, for the sake of brevity and to ensure clarity of the research methods, this study will examine Iconic Linkage in a monolingual environment without the added variables presented by the translation process. This notwithstanding, it should be borne in mind throughout that the discussion, principles and processes can just as easily be applied to the text production aspects of the translation process.

Defining "Quality" in User Guides

In discussions of user guides, it is not uncommon to see words such as *inadequate* and *adequate*, *good* and *bad* used. These terms are too vague for the purposes of any meaningful research and as such, it is necessary to select an appropriate and quantifiable criterion against which it is possible to measure the performance of user guides. Since user guides are effective only if users can use both them and the products easily, a logical measure of quality is the usability of the user guide.

To determine the effect of Iconic Linkage on the usability of user guides, it will be necessary to explore a range of areas to gain the requisite background knowledge. In *Chapter 2*, we examine the field of technical writing to understand the context of user guides and to gain an insight into how they are produced and what constitutes best practice. This chapter also looks at common evaluation methods.

To understand usability, it is necessary to examine the key components which affect it, i.e. humans and their cognitive abilities. *Chapter 3* examines users from a cognitive and problem-solving point of view and identifies the abilities and limitations of human cognition.

Building on this knowledge, *Chapter 4* takes the strengths and weaknesses of human cognition and seeks to find ways of incorporating this knowledge into the design of interfaces. In this case, the interface is the user guide, which is an interface between the user and the software. The chapter also introduces Iconic Linkage and describes it in detail whereby it is presented as a strategy for engineering interfaces to take into account the abilities and limitations of users.

Chapter 5 seeks to establish whether or not Iconic Linkage can improve the usability of user guides in practice. In order to do this, it is necessary to review various experimental methods, procedures and models before we can develop our own experimental model for assessing Iconic Linkage in a user guide.

Chapter 6 summarises the preceding chapters and draws conclusions on what has been learned as a result of the empirical study. This chapter also discusses possible avenues for further research in the area.

1.3 Anticipated Outcomes of Study

It is hoped that this study will provide clear and reliable empirical evidence to show whether or not Iconic Linkage can improve the usability of software user guides. By examining a broad range of subjects such as technical communication, cognitive psychology, cognetics and usability testing, it should be possible to gain a better insight into the production and translation of user guides as well as the way in which people use instructional texts. This study will also provide an insight into instructional design, usability, memory and learning as they relate to software user guides. Indeed, selected findings and results of this research have already been used in a number of conference presentations and publications relating to translation and technical communication (Byrne 2003a; 2003b).

Finally, as a result of this study, it is anticipated that we will have a new model for conducting usability reviews of user guides which can be applied to both monolingual and translated documents.

Chapter 2
SOFTWARE USER GUIDES

2.1 INTRODUCTION

This chapter deals with the genre of software user guides. It begins with a discussion of the field of technical writing, explaining what it is and what types of documents it produces. In addition, the chapter will also discuss software, software documentation, users and user guides. Finally, it examines in detail what are regarded as "best practices" in the production of user guides as well as ways in which the quality of user guides is assessed.

2.2 User Guides

This section discusses the genre of software user guides. In the context of technical documentation, several terms are used to describe what is essentially the same thing. Thus, it is not uncommon to see the terms "user guide", "user's guide", "manual" etc. in use. These terms are largely synonymous and for the purpose of this study, we will refer solely to user guides.

2.2.1 What is a User?

It would be easy and quite tempting to embark upon a discussion of user guides and describe linguistic features, style, technical and design strategies etc. without ever examining what we mean by "user." However, such an approach would provide only a partial picture of user guides and indeed it would leave us without a full understanding of the reasoning behind the features and strategies we would discuss. In order to understand user guides we must first understand users.

Weiss (1985:4) admits that any definition of user is difficult but nevertheless he endeavours to define a user as "a person more concerned with the *outcome* of data processing than with the *output*" [italics in original]. Weiss explains that users are "people who must be satisfied." They must be convinced that their "goals have been met and the advantages realized" (*ibid.*). This must be achieved, he continues, with acceptable effort and cost. Weiss continues to say

> ...users may become interested in the inner workings of computer technology. But this does not alter the basic idea: Users are people who want something bigger than, and outside of the particular device (*ibid.*).

It is clear from the preceding definition that users use software as a means to an end – a tool to help them do something else.

2.2.2 What is a User Guide?

Major defines a user guide as…

> a summary of all related resources […] written for everyone who might use these resources. It serves as a system of pointers to more detailed information, such as references manuals, tutorials, and standard operating procedures. (Major 1985:122)

At first glance, one would be forgiven for thinking that a user guide was little more than a directory or list of existing documentation. However, Major clarifies this when discussing documentation policy within a particular company. With an abundance of tutorials, how-to documents etc. a user guide was created in order to organise and arrange the existing documents. So rather than being a mere directory, Major explains that the user guide actually consists of many of these resources while providing references to others. He goes on to say that a user guide is a resource which not only points to all available documentation but also guides the reader through a general background and lays the foundations upon which "assumptions can be made about the level of comprehension readers have when they reach a given point in the total documentation picture" (*ibid.*).

Weiss (1985:4) provides a simpler definition of a user guide (or manual to use Weiss' terminology): "a user manual is – or should be – a tool that helps its readers get full benefit from the system." The guide, he maintains, is intended to compensate for the fact that software and information technology are often difficult and unfriendly to the user. This is also true by virtue of the fact that there is a limit to what can be learned autonomously and intuitively without assistance.

2.2.2.1 *The Function of User Guides*

The previous paragraphs have hinted at the functions of user guides. However, only provided a very cursory insight. According to Weiss (1985:16), the primary goal of user guides is to control the reader and the communicative action. Weiss maintains that

> ...to communicate well we must respect the independence and intelligence of the readers, but must not rely on them. [...] For user and operations manuals, the best strategy for writers is to adapt to the weaknesses in typical readers, to assume *control* of the communication. (*ibid.*)

Weiss is first to admit that any reference to controlling the audience or communication can raise strong ethical criticisms. However, he justifies this by saying that while we do not fully understand how people read or comprehend, we do have some knowledge about what distracts them or causes interference in the reading and comprehension processes. Thus, in removing sources of "noise and error" (*ibid.*) and things which we know will interfere with the correct and effective use of the guide we are in a sense assuming control over the reader and the communicative act.

If, for example, to quote Weiss a guide is little more than a collection of facts and pieces of knowledge, the effectiveness of the guide depends on how well the reader processes, sorts and assimilates the information. If, on the other hand, the guide is "engineered to suit the interests and abilities of the reader, then the user is to some extent prevented from misusing the material" (*ibid.*). In this regard it would, perhaps, be better to rephrase the goal of user guides and say that they should *guide* the reader and the communicative act by limiting what the reader can do with the user guide and limit the use of the guide to the intended one.

Other Functions

Although the function of user guides is to educate and guide the readers, Weiss (1985:12) divides this *instruction* function into a further three sub-functions which may be attributed to individual user guides to varying extents.

Tutorial

According to Weiss (*ibid.*), tutorials are "instructional materials intended to train neophyte users." Typically, such materials follow a simple task-orientated format aimed at helping absolute beginners take the first crucial steps in using a product. The subjects covered in tutorials tend to be basic, bottom-up concepts. Tutorials thus can be described as providing "learning support" as opposed to the purely procedural support provided by instructions.

Demonstration

Weiss defines demonstrations as "materials aimed at teaching a process or activity to a competent or experienced reader." Such materials literally teach users by showing them how to do something. Demonstrations show entire processes in a top-down manner because the users already have an understanding of the basics, what they need to know is how to put these basic steps together to complete a significant task.

Reference

This type of information is aimed at advanced users who, according to Weiss "know what they need to know." Rather than explaining why certain information is important and how it relates to other information, the reference material is simply a compressed version of key facts relating to the product.

Motivation

Often the people who read user guides are under no obligation to actually follow the instructions provided in a user guide. It is also the case that they may not even want to read the user guide in the first place. Schriver maintains:

> Bad experiences with documents influence not only our thinking about documents we've used but also our attitudes about documents we have not yet seen. People appear to generalize their bad experiences (Schriver 1997:4).

In such instances perhaps one of the most important functions of user guides (apart from pure instruction) is to motivate the reader, to make it seem worthwhile for the reader to read and follow the guide. Examples of such "motivating" guides would include the "*...For Dummies*" books published by *IDG books*, which take a range of different subjects and present them in an easy to follow way for people who, for whatever reason (often because the subject is perceived to be too difficult), may be reluctant to learn. With this in mind it is, according to Borowick (1996:176), important for the writer to explain the importance of the guide and the instructions it contains. Borowick maintains that a conversational tone, among other things, will encourage the readers to co-operate.

Borowick begins by saying that it is necessary for the writer to discuss at the very outset the "desire to achieve a common goal" (*ibid.*) or to express concern for the satisfaction and success of the reader as a user. The results and benefits should, therefore, be explained to the readers. The explanations or justifications for certain instructions must be clearly separated from the actual instructions. Borowick says this can be achieved using italics, parentheses or other "literary mechanical devices."

2.3 User Guides and Quality

Throughout the preceding sections we have looked at what user guides are, who they are aimed at and what they are used for. We have also looked at how user guides can be structured. We now need to examine how to determine the extent to which the user guide is effective – in other words, to find out how good the user guide is.

2.3.1 What Makes A Bad User Guide?

Unfortunately, many of the problems facing user guides are present even before the writer has put pen to paper so to speak. The mistakes and problems present before a user guide is written are, according to Weiss (1985:20) "nearly impossible to correct after the first complete version of the publication is drafted."

Price (1984:6) also acknowledges such problems and highlights the fact that there are certain obstacles placed in the path of good user guides even before writing begins. Price cites excessively tight schedules which limit the time available to produce a quality manual as one of the main problems. He also maintains that the failure to define the intended audience for the user guide is a critical error. The reason this happens, he maintains, can be out of simple ignorance of what goes into producing a user guide. Another reason, he claims, is the attitude of certain engineers and design team members towards users. Some engineers regard users as an annoyance. In practice, writers are faced with a situation where they must write to please the development team and so the attitude of the engineers becomes part of the user guide. Indeed, Price claims that some engineers even go so far as to ignore the user and their needs stating "If they don't understand this, they're not qualified to read it" (*ibid.*). This attitude is also reflected in what Price terms "poor design" in that no real thought is put into what the users need.

There are, according to Weiss (1985:18), four fundamental criteria for ensuring the effectiveness of user guides. Any errors on these levels will affect the quality and effectiveness of the user guide. Weiss presents the criteria as follows:

Availability
Perhaps the most obvious criterion – there must be documentation in order for it to be good. Weiss maintains that there is a surprising number of companies who do not produce user guides.

Suitability
User guides must be tailored to the specific needs of the users. Rather than write an enormous volume of biblical proportions which contains everything a user could ever need to know, separate documents should be produced specific to the needs of the different types of user.

Accessibility
As previously mentioned, a user guide should be more than just a presentation of information and facts. Just because the information is contained in a document does not mean users will know how to use it effectively; they may not even know when to use it. User guides need to be structured so that all of the information a user needs at a particular time is located together. This reduces the need for what Weiss calls "GOTOs" (GOTO is a programming command and refers in this instance to instructions to readers telling them to go to some other part of the document).

Readability
Weiss claims that, even when a user guide is suitable and accessible, a true indication of its quality is its readability. The language must be appropriate and easy to read and the document must be professionally edited to ensure proper style.

2.3.2 What Makes A Good User Guide?

Setting aside the organisational problems discussed in the last section, there are a range of areas which directly affect the quality and effectiveness of user guides. Traditionally, however, such issues as audience definition, document style and design were ignored and user guides were often produced as an afterthought. A turning point in the development of user guides came in 1978 when Apple Computer Inc. launched a new breed of user guide to accompany its new computer.

The following is an extract from a press release issued by Apple Computer Inc. to mark the launch of the guide:

> Created with foremost concern for the reader, the book assumes no prior background in programming or computers. Programming is explained in everyday English with no computer jargon used. Moreover, with scrupulous attention to detail, the book introduces the whole computer to the reader. Thus unlike programming manuals that solely teach a language, this book teaches a language in the context of the computer in which it will be executed.
>
> Another contrast with stereotyped programming manuals is the book's graphic illustration and **literary** style. Using a two-color process with text in black, significant information is highlighted in eye-catching green. Moreover, to illustrate displays, actual television displays are used to ensure the reader that observations on the television monitor will be the same as those within the book. Furthermore, the manual's informal, slightly humorous style, makes the book truly enjoyable to read. [emphasis my own] (Apple Inc. 1978)

One important point which should be made here is that the claim regarding the book's **literary** style should not be taken at face value. In this case, it does not mean literary in the traditional sense. Instead, this book used language devices and styles in stark contrast to previous documents with their dry, machine-like and "robotic" language.

This development set a new standard in the production of user guides and manufacturers soon realised that good user guides could actually win them customers (Price 1984:7). And so, terms like usability and design became part of the day-to-day vocabulary of technical writers and documentation departments as user guides came to be treated like devices (Weiss 1985:10-11). Companies learned that documentation needed to be well written, easy to understand, well laid out and presented, enjoyable and colourful. Of course there are countless factors which play a part in the quality of user guides and it would be impractical to discuss each and every one of them. However, it is possible to examine the relevant literature (Houghton-Alico 1985:59; D'Agenais & Carruthers 1985:48-50; Schriver 1997:263; Borowick 1996:132) on technical writing as well as a selection of contemporary user guides to gain an indication of "best practices" in this area. For the purposes of this study we will group factors according to the general areas of user guides they relate to, namely: *Appearance*, *Presentation*, *Structure* and *Language*.

2.3.2.1 Appearance

The appearance of user guides depends mainly on what kind of desktop publishing (DTP) processes have been used. Appearance includes anything from the arrangement of text on a page to the way a guide is bound – all of which affect the effectiveness and usability of a user guide. Houghton-Alico (1985:59) maintains that in order to produce quality software user guides we need to devote as much time and effort to format and design as we do to content and writing style.

Page Design and Layout

Apart from making a user guide more aesthetically pleasing, the way in which the pages of a user guide are designed and laid out plays a crucial role in how readers find and assimilate information and can even determine the environments in which the guide is used. According to Houghton-Alico (1985:59), each page should invite the reader to read the page, to become involved in the user guide.

The design of each page should, according to D'Agenais & Carruthers (1985:48-50) take the following criteria into account:

Simplicity
The design of the page should not be distracting or visually "busy." The information should be immediately apparent to the reader.

Retrievability
The page should have enough information on it to facilitate the immediate identification of the subject matter. For example, the document title or chapter name should be printed in the header or footer.

Flexibility
The design must be able to accommodate all variable data such as department names, people, etc. without any formatting changes.

Readability
Schriver (1997:263) maintains that the best length for a line of printed text is approximately 40-70 characters or 8-12 words as it is easier to read. A similar point is made by D'Agenais & Carruthers (1985:101) who say that a sentence should be between 10-15 words in length. The page should also have plenty of white space (see below).

Functionality
The page margins should allow for binding and double-sided printing and the headers/footers should use as little space as possible in order to maximise the amount of space available for the actual text of the user guide.

White Space

Perhaps one of the most important factors in page design is the relationship between printed matter and white space. Borowick (1996:132) defines white space as "any part of a page that is blank and used to separate ideas." This relationship is known as spacing and includes spaces between lines, paragraphs and margins.

Margins
Margins should be wide enough not just to facilitate binding (Austin & Dodd 1985:50) but to increase the amount of white space and "prevent the reader's eyes from running off the end of the page" (Borowick 1996:130). It is generally agreed that the page margins should be at least 1 inch on all sides with an additional 0.5 inch for the inside margin (Borowick 1996:130; Houghton-Alico 1985:59) although D'Agenais & Carruthers (1985:185) suggest a 0.7 inch margin.

Columns
White space can be increased by using a two column format where the left column is used for headings and the right is used for body text (Mancuso 1990:139). The following diagram illustrates this concept.

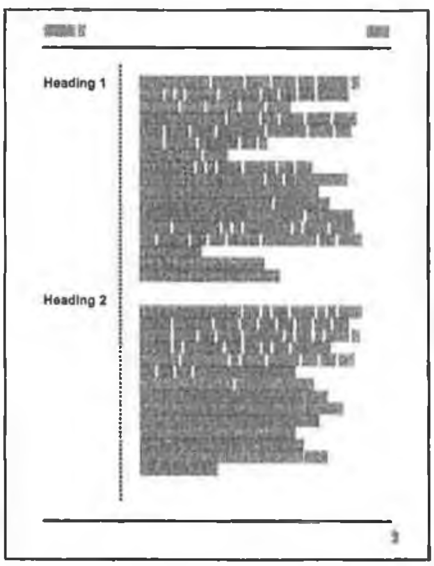


Figure 2: Two Column Page Layout

Paragraph Spacing

Paragraph spacing refers to the way separate paragraphs are presented as separate "chunks" of information using white space. According to Mancuso (1990:133) paragraphing and paragraph spacing reduce the amount of fatigue experienced by readers. As regards the actual amount of space between paragraphs, Mancuso (1990:139) recommends skipping lines. Houghton-Alico is more specific in this regard and says that the spacing between paragraphs should be 50% more than the space between lines.

Line Spacing

The spacing between lines of text is generally 2 points (see *Typography and Formatting* below below) larger than the size of the fonts used in the text. So, for example, if the font size is 10-point, the space between the lines will be 12-point. Mancuso (1990:139) recommends using one and a half or double line spacing between all lines of text. This is echoed by Borowick (1996:131) who states that single line spacing looks crowded and is difficult to read. One and a half spacing, according to Borowick (*ibid.*), allows the reader's eyes to "drop naturally at the end of each line to the beginning of the next line." Double spacing, he continues, is excessive and makes for uncomfortable reading.

Paper Size

One rather obvious factor affecting the amount of white space is the actual size of the paper being used. Quite simply, a larger page allows for more white space once the page margins have been incorporated and the pages have been bound. While standard paper sizes (e.g. DIN A4 or US Letter) are preferable, it is possible to use different sizes. However, using non-standard paper sizes can significantly increase production costs and may also affect how easy a user guide is to use (e.g. it is difficult to use a large and bulky guide while seated at a computer workstation or while working in a factory production hall).

Typography and Formatting

The fonts and formatting used on text in a user guide play an essential role in the effectiveness and usability of a user guide. The main requirements when choosing a font is that the fonts are clear and consistent (Austin & Dodd 1985:50; White 1996:204).

Fonts can be defined using three basic characteristics (White 1996:203):F

Font Type

Font type refers to the general design of the font. Fonts are divided into what are known as *serif* and *sans-serif* fonts. Serifs are small strokes or continuations at the top and bottom of individual characters or letters. Serif fonts, such as *Times New Roman* and *Bembo*, feature these small strokes; sans-serif fonts, such as *Arial* and *Century Gothic*, do not (see Figure 3).

Whereas sans-serif fonts are clearer, more visually striking and generally ideal for titles and headings, serif fonts are easier to read over a prolonged period of time and are suited for use on body text. This is because the serifs allow the reader's eye to follow the reading line better by leading from one letter to the next (D'Agenais & Carruthers 1985:185).

Font Styles

Font styles refer to variations or alternative forms of a font style such as *italics* or **bold**. Italics can be used to highlight definitions, to emphasize words, to denote trademarks or foreign words. Italics can also be used to highlight a word the first time it is used. Bold is used to emphasize words or to draw attention to certain information.

Font Size

Font size is measured in points. Sizes generally range from 5-point for small text right up to 72 for headlines. Standard text normally uses 10 or 12 point fonts.

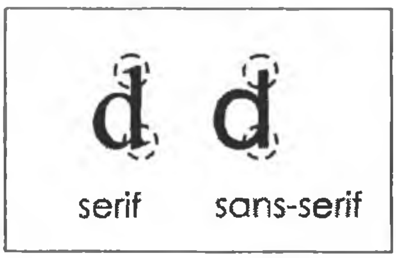


Figure 3: Serif and Sans-Serif Fonts

Text Alignment

The amount of white space and thus the readability of the text are affected by the way text is aligned on the page. There are four principle types of text alignment. *Centre alignment* is useful for headings and titles. Right alignment involves placing the end of each line of text flush along the right-hand margin. Conversely, left alignment involves the start of each line of text being placed flush along the left-hand margin.

The result is what is known as "ragged right" edge along the right-hand side of the text caused by the uneven line length (Mancuso 1990:144; D'Agenais & Carruthers 1985:185).

Justified alignment involves "stretching" each line so that the start of each line is placed flush along the left-hand margin and the end of each line is placed flush along the right-hand margin. This type of alignment is achieved by adding additional spaces between certain words in order to pad the length of shorter lines.

This decision to use one type of alignment over another, in particular ragged-right versus justified is not without its problems. Where some studies, such as Gregory & Poulton (1970), have shown that may be more difficult to read, others such as Fabrizio, Kaplan & Teal (1967) and Hartley & Burnhill (1971) found no difference in reading speed or comprehension between ragged-right and justified text.

While justified text looks neater on a page, it is more difficult to read than ragged-right text if there are "rivers" of white space running down the page. The problem here is that the additional and often unpredictable spaces added to the lines make it difficult for the eye to proceed at a constant rate along the line of text. Each time an extra long space is encountered, the eye pauses briefly. Such pauses or hesitations are known as "fixations." In contrast, the even spacing between words in left aligned text eliminates this and the uneven line length adds variety for the reader and makes the text less monotonous. It is widely recommended that body text be left aligned (D'Agenais & Carruthers 1985:185).

2.3.2.2 *Delivery*

Delivery is used to refer to the physical means of producing and distributing user guides. The way the user guide is delivered affects the way the guide is used, how frequently it is used as well as how easy it is to use.

Paper Type

The type of paper is important particularly in the case of larger user guides where pages are printed on both sides in order to reduce the overall size of the guide. If the weight of the paper (measured in grammes per square metre – gsm) is too low, the paper will be too transparent and text from the reverse side of the page will show through. In addition, depending on where the user guide is to be used, a heavy paper may be necessary, for example, user guides intended for use in a workshop or garage.

Binding

In addition to impacting on the margin sizes and consequently the amount of white space available on each page, the type of binding used on a user guide affects the usability of the guide. For example, a user guide which is only used occasionally or in an office context could be *perfect* bound whereby the pages are glued together at the edge and covered with a heavier sheet of paper or card. User guides which contain information sheets or which will be updated regularly would be best served by *loose leaf, ring binding*. Another type of binding is *easel binding* which bends in the middle and can be stood on a desk or bench (D'Agenais & Carruthers 1985:46).

Tabs and Tab Dividers

Tabs and tab dividers are an important part of binders (D'Agenais & Carruthers 1985:226) and are used to make navigating through a document easier. Tab dividers can either be numbered or they can feature the names of chapters or sections.

2.3.2.3 *Presentation*

In addition to textual information illustrations and pictures of the software interface can also be used in user guides. These pictures are known as *screenshots* or *screen grabs* and an example is given in Figure 4 below.

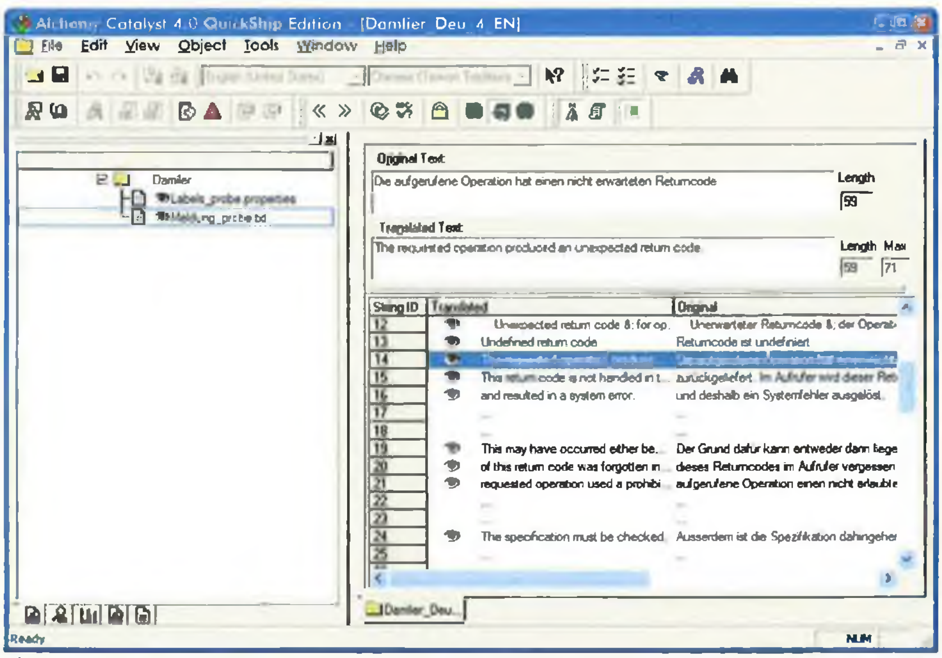


Figure 4: Screenshot from Alchemy Catalyst 4.0 QuickShip (© 2002 Alchemy Software)

However, the range of visual devices is not restricted to screenshots. Other devices include tables, graphics, graphs, charts and diagrams.

Tables

Tables are a useful way of presenting information in a text to make the information clear and accessible. They are frequently used because they present information without the need for any interpretation on the part of the writer (Borowick 1996:112). Indeed, White (1996:220) regards tables as data cross-referencing devices whose value lies in the speed with which data is "perceived and configured." Tables can be used for both numerical data and textual information and provide readers with "latitude to analyze and understand their meaning" (Borowick *ibid.*).

Borowick also provides the following guidelines for the creation and use of tables:

- Put the table number with a caption near the top row of the table. White (1996:233) also states that the caption should be as brief and informative as possible.
- Label the top of each column and the left side of each row.
- If the units of measure are all the same in tables containing numerical data, the units should be placed with the caption of the table. For example, "Length, cm.". Alternatively, Borowick suggests placing the units in parentheses in the column or row headings.
- In a table containing numerical data, line up all decimal points in each column. This idea is also proposed by White (1996:223).
- Any item which requires detailed explanation should be annotated with a superscript number, e.g. "Volume[1]". The explanation should then be placed immediately after the table.

White (*ibid.*) also maintains that the table should be introduced in the body of the text, preferably in the preceding paragraph, for example "Table 2.3 below illustrates the statistics for…". A table should not come before its first reference in the body of the text.

Graphics

The term "graphics" means graphs, charts, pictures, photographs, icons, diagrams and drawings. The target audience is a key factor in deciding what type of graphic to use.

Graphics are an aid to communicating ideas and concepts clearly and quickly rather than using paragraphs of text explaining the same concept. When it comes to actually implementing graphics in a user guide, there are a number of generic guidelines which should be considered:

- Graphics must be referenced sequentially in a document (Borowick 1996:102)
- A graphic should appear on the same page as its reference in the body of the text. A graphic should, according to Borowick (*ibid.*) never come before its first reference in the text. Sometimes, however, documents may have all graphics collected at the end of the document. Nevertheless, they should be numbered, captioned and referenced.

Graphs and Charts

Graphs are used to depict trends or relationships between variables. They are a visual representation of information and are a good way of conveying information quickly.

Photographs

While photographs are the most realistic way of representing an object, their use is not always advisable because, according to White (1996:236) it is "difficult to be selective about the visual information you wish to present". For instance, if a user's attention is to be drawn to a particular part of a computer's hardware (e.g. the motherboard), a photo of the entire internal mechanism of the computer will leave the reader searching for the part in question. A drawing or diagram of the particular area would be more effective because, depending on how the diagram is drawn, it is possible to focus attention onto the specific part while eliminating distracting visual information.

Photos also raise the issue of print quality and printing costs as they must be of a sufficiently high quality in order to ensure that they are clear and effective. However, the use of high-quality photos also raises the cost of printing the documents.

Drawings and Diagrams

Drawings and diagrams are visual reconstructions of an object and may be representational or abstract. They help the reader to visualise physical objects.

2.3.2.4 *Structure*

According to Weiss (1985:50), user guides are structured in that they represent a top-down approach to a particular task, e.g. providing information on a software application. By this we mean the user guide starts with the "big picture", the largest possible overview and then progressively and systematically adds more and more detailed information. While it is generally held that this top-down approach involves breaking concepts into smaller and smaller ideas (cf. D'Agenais & Carruthers 1985:68-9) – a process known as decomposition (Weiss 1985:50) – it is, in fact, only a small part of the structured approach to user guide design.

First and foremost, a user guide needs to provide a broad overview in order to explain how the various constituent ideas and tasks relate to each other. This can be explained as follows: using the example of a piece of website management software, the big picture is creating a website but in order to do this we need to create the web pages. We then need to create directories in which to store the pages and, once we have done this, we then need to know how to upload the pages and directories to the web server.

As a result, we can see that we need to create the directories in order to store our web pages effectively and to allow navigation but these pages and directories cannot become a website unless we know how to upload them. Similarly, we cannot upload a site if it is not organised into directories. Neither is there any point in uploading directories if they do not contain web pages. So in addition to breaking ideas down into smaller tasks, we need to tell readers how these tasks all relate to each other. Thus, a user guide will generally consist of many small units or modules, all of which are connected in a way that will make it clear to readers what they are learning and why.

Modules

Weiss (1985:52) defines a module as a "small, independent, functional entity, a component of some larger entity". He continues by describing modules under the following headings:

Modules Are Functional

Modules do something, they perform some task. What is more, they perform a complete task with a clear beginning and end.

Modules Are Independent

Modules do not depend on their context. Since there is a clear beginning and end, the module can function in isolation and may even perform that function in more than one situation (e.g. a module explaining how to save a file under a different name in Microsoft Excel would be equally effective at explaining how to copy a document to another location using *Microsoft Word*). Thus, modules can, according to Weiss (*ibid.*), become part of a "library of reusable modules". And so these modules eventually resemble a set of stock modules which can be picked and mixed by writers.

Modules Are Small

As already stated, modules perform only one function (even though this function can be performed in a number of contexts). But this does not give us a clear understanding of how small a module should be. It could be argued that it is impossible to say with any degree of certainty how small is small. Weiss (*ibid.*) posits the notion that as modules get larger, they begin to incorporate more functions and begin to lose cohesiveness. On the other hand, as they get smaller, the links between modules become more complicated. So in the absence of any real alternative, we find ourselves faced with a balancing act of "not too much, but just enough". A practical approach would be to concentrate on one "task" as opposed to a function.

Distribution and Organisation of Modules

A table of contents (TOC) is a map of the document. It is an outline of the document including the major and minor topics. According to Price (1984:65), a TOC should at the very least include chapter titles and the first level of headings from each chapter as well as the page numbers for each title and heading. However, depending on the exact content of each chapter, additional levels of headings may be necessary.

Price (1984:65-66) provides a number of guidelines which are of use not only to writers but also to translators. The first of these is that chapter titles and level 1 headings should make sense to beginners. They are generally quite broad, for example, "Installing...", "Configuring...", "Using the Software", "Solving Problems" etc. However, the headings within chapters become more specific and tell readers what they can do or what the section can do for them. One way of reinforcing the notion that a reader can do or achieve something is to use verbs in headings as opposed to "a bunch of nouns, which look like a list of topics to study" (Price 1984:65). Price also suggests phrasing headings as questions as a way of making them more interesting and informative. It could be argued that this is because the questions may actually reflect questions the reader may be asking. An interesting examination of headings is provided by Gerzymisch-Arbogast (1993) who argues that texts are composed of information that is *given* or *new* relative to the reader. The relative amounts of this information and the way it is sequenced are, she says, dependent on the author-reader contract governing a particular document and must be changed to suit the needs of a new audience. Thus, in an English document, headings generally contain either given information or a combination of given/new information.

In addition to making headings informative and clear, the very nature of the TOC which is generated from the actual headings used in the text places constraints on the length of titles. Short titles, in addition to looking neater on a page, are actually easier to read (Price 1984:67). A reader should not have to scan a heading a number of times in order to elicit the information. Price (*ibid.*) recommends that titles should not take up a full line.

Overviews and Summaries

As readers progress through a user guide or even when they dip into a user guide to read a chapter, it is important to tell them what they will find in a particular chapter or large section. This helps them to decide whether the particular section is what they are looking for. Overviews can also help readers absorb, understand, learn, and remember information more easily (Foss *et al.* 1981).

Price (1984:72) and D'Agenais and Carruthers (1985:90) state that every chapter and large section should have an overview. Consistent with the motivational function described on page 20, overviews explain to readers why they should read a particular chapter or section, what they will be able to do afterwards and what they will be able to achieve with this knowledge.

In general, a typical overview will explain what readers have learned so far (providing they have been reading the user guide in sequence) and how this particular chapter builds on previous chapters. The overview tells the reader what will and will not be covered in the chapter and provides a broad idea of the subjects which will crop up. An overview can also suggest strategies as to how different users should use the chapter or document, for example, whether to skip certain sections or whether to read another chapter first. It may also provide references to other sources of information.

Reassuring the Reader

Another way of organising information into manageable sections is to provide regular "breaks" for readers where they are given time to rest, absorb what they have just learned or even just to have a clear point in the text at which they can close the book and still feel they have achieved something worthwhile. Such breaks may come in the form of a congratulatory remark such as "*Well done, you've just learned how to XXX. Why not take a few moments to try out your new skills?*" or even a suggestion that they make a cup of coffee and relax for a moment (Price 1984:91).

More difficult information requires a different type of break which is provided before the readers are actually confronted by the information. If, to quote Price (1984:91), a potentially worrying message is about to appear or the chapter is about to deal with complicated information, the reader should be reassured and told not to worry. Price suggests that the text should admit that a particular message is worrying or that information is complicated but indicate that the reader will be able to manage it.

2.3.2.5 Language

The actual language used in a user guide is probably the most critical factor in determining its quality and effectiveness. Indeed, the text provides the sensory stimulus which conveys the information to the readers' brains. However, just like language itself, the factors which govern how effectively the text is used are equally vast. There are myriad guidelines, rules and regulations such as EN 62079 which are aimed at improving the standard of language but it would be impractical to discuss them all here. Rather, we can group a number of the guidelines into the following supersets: *clarity and word choice*, *sentences*, *style*, *verbs* and *text mechanics*.

Clarity and Word Choice

Reminiscent of the old adage "less is more", a commonly held tenet of technical writing is that texts should be as brief and concise as possible and writers (not to mention translators) should eschew verbosity (D'Agenais & Carruthers 1985:100-101; Weiss 1985:148-9, 152). According to Weiss (1985:148) the most frequent "offenders" with regard to verbosity are what he calls "smothered verbs". A smothered verb, also known as a nominalisation, is a verb that has been converted into a noun, e.g. "they conducted an investigation" instead of "they investigated". Nominalisations involve using a phrase where a single word would have sufficed and also encourage the use of unwieldy passive constructions.

Conversely, however, it is possible to be overly concise and compress text to such an extent that it becomes incomprehensible or ambiguous. The notion that text can become ambiguous as a result of excessive compression is echoed by Ramey (1989) who describes the incidence of Escher effects in texts. Escher effects - named after Escher's famous two faces / one glass picture – result in a phrase or piece of text having two or more possible meanings and force readers to truly study the text in order to ascertain or deduce which meaning of the text is the intended one. The following examples illustrate Escher effects in text:

- input mode
- operating system file specification rules
- programming error messages

Each of these examples can have a number of possible interpretations. Taking the first example we can see that "input" can be read either as a verb or as a noun. So it is conceivable that one reader will regard "input mode" as a command – that the reader is required to input or specify the mode. Meanwhile, another reader may regard "input mode" as a state, that "input" modifies or qualifies "mode".

The specific type of words used in a text can play an important role in its quality. D'Agenais & Carruthers (1985:106) suggest that positive words be used instead of negative words because, presumably, negative words have an undesirable effect on readers. The authors give the following example which is admittedly a little contrived but which does illustrate the point:

- Lock the door when you leave.
- Don't neglect to lock the door when you leave.

D'Agenais & Carruthers (*ibid.*) go on to say that words can be used to smooth the transition from idea to idea, sentence to sentence and paragraph to paragraph. The purpose of this is to avoid abrupt changes which can leave readers wondering where to go next. This idea is consistent with the theory behind the Müller-Lyer Illusion (Coe 1996:29). Figure 5 shows two lines, A and B. Both of these lines are of equal length, and each has arrowhead tails: on line A they point back over the line and on line B they point away from the line.

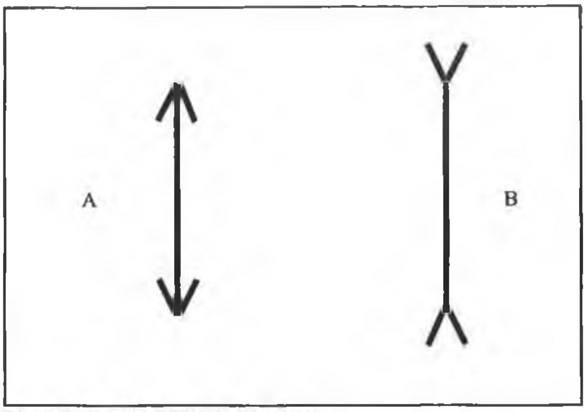


Figure 5: Müller-Lyer Illusion

Although both of these lines are the same length, the way our brains perceive the lines tricks us into thinking that line B is longer than line A. The reason for this is that the arrowhead tails on line A direct our attention back onto the line while the arrowhead tails on line B direct our attention away from the line. Similarly, textual transitions provide a link between ideas or stretches of text and offer a pointer directing the reader where to go next (Coe 1996:29).

Returning to more general aspects of word choice, it is, perhaps, useful to remember that a key goal of user guides is that they should present information in a simple manner. Simplicity of language can be obscured by a number of word choice factors: jargon, euphemisms, neologisms and abbreviations / acronyms.

Jargon

Each and every discipline, be it biology, precision engineering, electronics or meteorology has its own vocabulary of specialised terminology. This terminology is frequently referred to as jargon (White 1996:191; Mancuso 1990:186). Indeed, specialised terminology is essential in order to avoid ambiguity and to accurately communicate ideas and concepts. However, this terminology can also be an irritation and hindrance when misused (White 1996:192). The problem is, according to Mancuso (*ibid.*) that "experts use too much jargon in documents meant for less well informed audiences". The general consensus is that jargon should be used in a way that is appropriate to the abilities and level of knowledge of the audience (Mancuso 1990:186-7; White 1996:192). Where it is essential or unavoidable that jargon be used, the specialised terms should be properly defined (Mancuso 1990:186).

Euphemisms

Euphemisms are figures of speech which are used to describe things using milder, less unpleasant terms. They are generally used to soften or lessen the impact of harsh or unpleasant words or ideas. Euphemisms are frequently longer words or phrases and their meaning or relation to the actual object or action being referred to is less than obvious.

The problem with euphemisms is that while they are often quite clever, creative, linguistically interesting and occasionally amusing, they obscure meaning, confuse readers and generally make the text less accessible. In addition, because of their size, they make the text longer and more cluttered (Mancuso 1990:191).

Neologisms

Neologisms are, according to Mancuso (1990:197), the work of "arrogant" authors who like to create new words. Mancuso continues by saying that these newly created words are generally only understood by the author and a few others and they confound most readers. Admittedly, such a view is quite extreme and occasionally neologisms are necessary; they should, however, be used sparingly.

Acronyms and Abbreviations

Acronyms and abbreviations can affect the clarity and accessibility of a text in much the same way as jargon. Although many computer-related acronyms and abbreviations are becoming more widely known than they used to be (Mancuso *ibid.*), many are not yet in common usage. Thus, according to D'Agenais & Carruthers (1985:109), those that are not commonplace and understood by everyone should be explained. A popular way of dealing with acronyms and abbreviations is to use a glossary which explains them (Mancuso 1990:197; D'Agenais & Carruthers 1985:109).

Of course, clarity can also be affected by the ambiguous use of "ordinary" words. Ambiguity usually arises, according to White (1996:190), as a result of one or more of the following problems:

Improper word choice

- Using ambiguous words which can have more than one meaning in a particular context.

Unclear pronoun reference

- Pronouns must co-refer only with the noun phrase intended by the writer.

Squinting modification

- Sometimes a word can modify the phrase preceding it to give one meaning but also modify the phrase following it to give a different meaning.

Ambiguous relationships

- Using co-ordinating conjunctions such as "and" when a subordinate relationship is intended.

Sentences

> If words represent the colors of the palette available to the writer, sentences are the lines that create shapes in a composition (Houghton-Alico 1985:54).

Having looked at a number of factors relating to word choice and clarity, the next logical step is to look at how sentences affect the quality of a user guide. In line with our previous discussion of why texts should be concise, the issue of repetition and redundancy is worth examining. Firstly, we need to distinguish between repetition and redundancy. Repetition involves repeating words and phrases throughout a document in order to reinforce information, reiterate product benefits or to get readers to do or remember something. There is a definite purpose to repetition – perhaps merely to assist in the habit formation process (Raskin 2000:18-21). Redundancy, on the other hand, is "stated or implied repetition with no purpose" (Mancuso 1990:202). Redundancy can take the form of superfluous adverbs, hedge words, unnecessary emphasis or repeating information in a different form.

The flow of information in sentences is also of great importance with regard to the readability of the text. Indeed, Weiss (1985:150) argues that "the secret of the readable sentence is that the 'payload' of the sentence [...] is at the end". The payload is essentially the most important part or "nugget" of information the author wants to convey using the sentence. The reason why the payload should be at the end is, according to Weiss (*ibid.*), that the last part of the sentence is the best remembered by readers. Similarly, in the case of instructions, a cause-effect format should be adopted (SAP AG 1997:4ff). Accordingly we would, for example, rewrite the following sentence:

The tab marked Properties allows users to configure the modem's settings.

as

To configure the modem settings, click the *Properties* tab.

Parallelism

Parallelism is a phenomenon which is widely recognised as a fundamental issue in sentence structure (D'Agenais & Carruthers 1985:104; Mancuso 1990:231; White 1996:182). Essentially, parallelism means that parts of a sentence which are similar, or parallel, in meaning should be parallel in structure. Parallel constructions can also be described as instances where two or more groups of words share the same pattern (White 1996:182). Thus, we can see that parallelism can occur on both a sentence level and on a sub-sentence level. The following sentences illustrate parallelism.

If you want to open a file, click Open.

If you want to close a file, click Close.

Parallelism can also occur in lists as shown below:

To connect to the Internet you will need:

- a modem to connect to your PC
- drivers for your modem
- a telephone line
- a dial-up account from an ISP

When there is a lack of parallelism, some of the grammatical elements in a sentence do not balance with the other elements in the sentence or another sentence. What makes this undesirable, apart from potential grammatical errors, is that it distracts the reader and prevents the message from being read quickly and clearly (Mancuso 1990:232).

Returning to the examples of parallel constructions given above, we can illustrate how a lack of parallelism can affect the clarity and readability of a section of text. What were once clear sentences, become the following confusing examples:

If you wont to open a file, click Open.

The Close button should be pressed to close a file.

To connect to the Internet you will need:

- a modem to connect to your PC
- drivers for your modem
- a telephone line must be available
- also, contact an ISP to get a dial-up account

Parallelism is not just important in avoiding grammatical and comprehension problems, it is also very useful in reinforcing ideas and learning. The grammatical symmetry of parallelisms helps readers remember information more easily (White 1996:183). The notion of parallelism is closely related to that of Iconic Linkage which we will discuss in Chapter 4.

Style

When we speak of style, we really mean the overall tone of the text and how authors express themselves – essentially, how the authors relate to their readers.

It is widely acknowledged (D'Agenais & Carruthers 1985:104; Mancuso 1990:149; Davis 1992:11) that a conversational style is the best approach when writing user guides. Mancuso (*ibid.*) ventures by way of an explanation that the way we normally write is generally unsuitable for explaining ideas. When we explain ideas orally, we are concise, to the point and we avoid awkward or complicated constructions. Indeed, D'Agenais & Carruthers (*ibid.*) maintain that most people communicate better when they are speaking than when they are writing. A possible reason for this is, according to the authors, that people tend to "write to impress rather than to express". Mancuso and D'Agenais & Carruthers agree that the best way of avoiding "stilted prose, using passive voice verbs and other awkward constructions" (Mancuso 1990:149) is to explain things orally or in speaking mode rather than in writing mode.

Using a conversational style does not, however, give authors free rein to use slang, to be excessively informal or to be imprecise or ambiguous. While oral communication has the benefit of instant feedback from the receiver's reactions, written communication does not have this aid and so the potential for misunderstanding must be minimised.

Verbs

Verbs are the engines of sentences – they make the sentences meaningful and make a text more than just a list of words. The way in which verbs are used affects the way the text works and how easily the reader assimilates information. We can categorise our examination of verbs as follows:

- Strong / weak verbs
- Active / passive voice
- Imperatives
- Compound verbs

Strong and Weak Verbs

The differentiation between strong and weak verbs can be quite subjective and is rather elusive. It would, perhaps, be easier to define the two terms using a number of examples of strong and weak verbs. Mancuso (1990:174) suggests that strong verbs might include *weld*, *singe*, *salivate*, *bulldoze* and *inject*. Weak verbs, he continues, include the various forms of the verb *to be* and the verbs *do*, *make*, *provide* and *include*. Strong verbs, he maintains create images; they add a sense of action to a text. On the other hand, weak verbs say little, if anything and result in the reader having to spend more time "deciphering meaning rather than reading it" (*ibid.*).

From the examples given below, we can see that strong verbs are those that actually reflect the function or action in question. The following sentence is rewritten to illustrate examples of strong and weak verbs:

The function of the hard disk is to allow you to store data.

The hard disk stores data.

The benefit of using strong verbs is that it allows readers to understand information more quickly. Additionally, as can be seen in above example, strong verbs allow for more concise constructions.

Nominalisations, i.e. verbs that have been transformed into nouns, are just as unhelpful as weak verbs in that they obscure meaning and add to the workload of readers. An example of this would be as follows:

The setup program results in an update of the registry.

If we remove the nominalisation, we get the following:

The setup program updates the registry.

Active and Passive Voice

The terms "active" and "passive" voice are old metaphors for certain grammatical constructions. Active voice constructions contain subjects that "do something". These constructions have positive connotations of action, dynamism, energy and determination (White 1996:181). Passive voice constructions, on the other hand, contain subjects that do not do anything. These constructions have the opposite connotations to active voice constructions.

The passive voice is typified by the following characteristics:

- The subject is acted upon.
- The predicate generally contains an auxiliary verb that is in the form of to be.
- The sentence contains a prepositional phrase

While it may be helpful to switch between active and passive voice in order to emphasise either the subject or the logical object (White 1996:182), it is widely held that the passive voice interferes with the clarity of sentences (White *ibid.*; Mancuso 1990:156-171; D'Agenais & Carruthers 1985:102-3). It is also more difficult for readers to understand the sentence because of problems with identifying the actor and also because of delayed meaning (Mancuso 1990:166-7).

Imperatives

Using the active voice in conjunction with the imperative mood is an important strategy in procedural texts where the reader is required to either perform certain tasks or refrain from carrying out certain actions. In contrast to constructions that do not use the imperative, there is no confusion as to who is to carry out the task because the second person pronoun *you* is implicit (Price 1984:103). Take, for example, the following sentence:

The necessary drivers must be installed on the PC.

From this sentence it is not clear who is supposed to perform the task. Is it the reader or is it someone from the IT department? It would be better to phrase the sentence as follows:

Install the necessary drivers on the PC.

2.4 Assessing User Guide Quality

The evaluation of user guides takes place on a number of levels. While it would be tempting just to test a user guide to see that it accurately reflects the software, such an approach would provide only a partial picture. As we have seen in preceding sections, the design and layout of a user guide as well as the way it is written are important factors in producing a user guide. In the following sections we will examine several methods for assessing user guides.

2.4.1 Readability

In addition to ensuring that writing is clear, consistent and concise, readability testing also indicates whether the text is at the correct level for the intended audience. There are numerous methods for measuring readability including the *Flesch Readability Test*, the *Lensear Write Formula*, the *Fog Index*, *Fry's Readability Graph* and the *Clear River Test*.

Most of these methods involve selecting a sample of text between 100-200 words in length. Each of the methods mentioned above express readability in terms of the proportion of various features such as syllables, monosyllabic words etc., average sentence length etc. Methods such as the Fog Index regard words with more than three syllables as "difficult" while words with less than three syllables are regarded as "easy".

The Flesch Readability Test examines readability as a relationship between the average sentence length and the average word length; the shorter the sentence and the shorter the words, the more readable the text.

The Fog Index identifies easy words in a text, i.e. words with one or two syllables, and calculates readability as a function of the average sentence length and the percentage of "hard" words, i.e. three or more syllables. Readability is then expressed in terms of the number of years schooling needed to read the text with ease. Similarly, the Lensear Write Formula calculates readability on the basis of the proportion of monosyllabic words in a text.

Like the Fog Index, Fry's Readability Graph expresses readability in terms of the level of schooling needed in order to read a text with ease. Taking a sample of 100 words, it calculates the average number of syllables per sentence and the number of sentences per 100 words before expressing readability as a function of the two.

The Clear River Test combines several of the features of the preceding methods and analyses readability in terms of the number of words per sentence, per paragraph, per punctuational pause and the number of syllables per 100 words.

These tests can prove very useful in providing an overview of how effective a user guide is in terms of readability but they do not explain why a user guide is ineffective despite being readable. Is the text poor because of the register used? Does it contain too much jargon? Are concepts not explained clearly? Does the text contain ungrammatical constructions? Indeed, George Klare, a leading academic in the field of readability evaluation formulae concedes that readability assessments are of limited use in assessing computer documentation and that in some cases, such methods were not even designed for use on such texts (Klare 2000:2-3). It is clear that in order to pinpoint precisely what errors contribute to a text's under-performance, we need to find a more comprehensive evaluation method.

2.4.2 Usability

Another approach to determining the effectiveness of a user guide is to establish how easy it is to use, i.e. its usability. In contrast to readability assessment methods which examine linguistic and technical features from the point of view of the text, usability introduces a new element into the equation, i.e. users. Usability assessment evaluates linguistic and technical features such as those described in previous sections and assesses the sum total of all of their contributions from the point of view of the user. Instead of considering only the readability of text or whether the style is appropriate, usability is concerned with the ease with which users (readers) can access and assimilate information and then use it to complete their intended tasks, i.e. use the software.

A simple way of testing the usability of a user guide is to gather a group of people who reflect the actual audience for the user guide and have this group use the software on the basis of the user guide. The purpose here is to see where the readers succeed and where they go wrong, where they have difficulties and where they need more help. Usability testing of this type (as defined in technical writing literature) tests the user guide for logic gaps and inadequate clarity. It determines whether readers can actually use the user guide effectively and efficiently and whether users actually learn from it.

Usability is a central element of what is known as Human-Computer Interaction (HCI). This area is concerned with examining the interactions between humans and computer systems (e.g. software). The description of usability provided in the preceding paragraphs is admittedly rather simplified and rudimentary. In order to fully understand usability, it is necessary to understand the primary component of these interactions, i.e. humans. In the following chapter we will examine usability from the point of view of users and discover the mechanisms that must be understood and accommodated to ensure usability.

2.5 Conclusions

The preceding sections have sought to introduce the genre of software user guides and place it within the overall context of technical communication. It is clear from this that user guides are one single product of technical communication, yet they are, perhaps, one of the most visible products. It could be argued that they are one of the crucial types of technical document because they are instrumental in allowing new users to learn how to use new software. We have seen that the perceived ease of learning as facilitated by user guides can be a decisive commercial factor for software products.

This chapter also examined in detail the components of user guides. From this it emerged that a "good" user guide is more than just a collection of clearly phrased instructions or a repository of all information relating to a piece of software. Rather, user guides should ideally contain information that is targeted at the needs of the users and what they want to do. They present users with the knowledge they need in order to perform a task, when they need to perform it. Information is "fed" to users in a logical and timely way.

Beyond the purely stylistic and content-related issues, a range of other factors such as layout, typography, presentation, structure etc. influence the effectiveness of user guides. All of these factors paint a more holistic picture of the nature of user guides than that which is reflected in the methods commonly used to assess the quality of user guides. Readability tests such as the *Flesch Readability Test*, the *Fog Index* or the *Clear River Test* coupled with technical accuracy checks merely assess a small part of user guides.

Line spacing, white space and information chunking point to some form of understanding of how humans read text and perceive information. Simplicity of language, clear instructions, the use of parallel structures and active verbs and the avoidance of euphemisms etc., all draw on characteristics of the way humans decode, understand and absorb information. It is clear, therefore, that we need to examine these factors from the point of view of the person reading the user guide.

One area raised in the chapter which is worthy of further investigation is that of usability and usability testing. Unlike readability testing, usability testing seeks to understand all of the factors that influence how well users can use a user guide. Usability testing adopts a suitably broad approach which centres on the reader and it will give us a deeper understanding of why some user guides are easier to use than others.

In the following chapter, we will begin our examination by defining usability and discussing its importance for users. We will then look at the processes and systems that are called into play when we read a user guide. By understanding readers' cognitive abilities, preferences and limitations, we can begin to identify those aspects of user guides that facilitate the transfer and assimilation of knowledge necessary to use software -- the stated purpose of user guides. Thus, any discussion of usability requires a thorough understanding of the human cognitive system and its processes.

Chapter 3
HUMAN COGNITION & USABILITY: UNDERSTANDING USERS

3.1 INTRODUCTION

As stated in the last chapter, usability, the measure of how easily and effectively people can use something (in this case, a user guide), is a central component of *human-computer interaction* (HCI). In this chapter, we will look at usability and the interaction between humans and computers in detail. If usability refers to the extent to which people find something easy to use, to understand usability we must first understand the users who are the ultimate judges of usability. We will examine ways of modelling the human cognitive system and discuss how it works. The sensory, cognitive and learning processes will be examined as well as the way humans remember information. The various processes involved in assimilating and interpreting information will also be explained.

3.2 USABILITY

> "A computer shall not waste your time or require you to do more work than is strictly necessary" (Raskin 2000:6)

In Chapter 2 we discussed how making user guides more accessible and usable is a primary objective for technical communicators. It would be easy to produce a simple working definition of usability such as "ease of use". However, such a definition by no means explains the true nature of usability and the factors affecting it.

Another common fallacy is to somehow confuse usability and usefulness. While ostensibly related, they are poles apart in terms of their relationship to products. Usefulness refers to the potential uses users can find for something whereas usability refers to how well users can use it (Landauer 1995:4; Ehn & Löwgren 1997:301; Dumas & Redish 1993:5).

However, defining usability as a measure of how well users can use something is a slight over-simplification. In the ISO 9241-11 standard "Ergonomic requirements for office work with visual display terminals", usability is defined 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.

In other words, usability refers to how well a given user will perform a given task at a given time. There are difficulties, however, in this definition with regard to the phrase "specified users". A number of theorists maintain that attempting to define the concept of a user is highly problematic and possibly even futile (Ehn & Löwgren 1997:299; Bannon 1991:26-27). This is because there may be a vast number of different people, all with different backgrounds, knowledge, abilities, skills and patterns of use who will all be using the product. It would be prohibitively difficult to profile each type of user in terms of the aforementioned criteria. In view of this, Ehn & Löwgren (*ibid.*) propose that our focus should be on the situation of use, i.e. where, when and how the user uses the product.

Dumas & Redish provide a definition which is less specific than the ISO 9241 definition given above but which nonetheless provides additional insight. According to their definition, "usability means that the *people who use the product* can do so *quickly and easily* to accomplish *their own tasks* [emphasis in original] (1999:4). Here, the crucial factor is the fact that users are using the product to perform *another* task. The use of the product is secondary to a user's true intention. We can see, therefore, that usability does not depend on the product *per se*, but rather on the people who use it. A usable product is one which is appropriate to the **tasks** users want to carry out. Indeed, according to Faulkner (1998:7) "the very best systems and the very best interfaces will be overlooked entirely by the user" and ideally, all the user should see is the task and not the system.

Dumas & Redish (1999:4-6) examine the relationship between usability and users under the following headings:

- Usability means focussing on users
- People use products to be productive
- Users are busy
- Users decide how usable a product is

Usability Means Focussing On Users
In order to make a usable product it is vital to understand real users. People such as managers and developers do not represent real users.

People Use Products To Be Productive
Software products are tools which people use in order to do something else. People judge the product on the basis of the time it takes them to do something, the number of steps they have to perform and how successful they are in doing the task. The aim is to make products so easy to use that users can perform their tasks more quickly.

Users Are Busy
The usability of products is gauged by users in terms of how quickly they can get the product to do something. A product may have precisely the functionality a user needs to perform a task but if the function cannot be accessed or used within the time the user is prepared to devote to the task, it will be useless. This idea is virtually identical to Landauer's distinction between usefulness and usability discussed above.

Users Decide How Usable A Product Is
Regardless of how well developers, managers or marketing people think something is designed, the ultimate judge of usability is the users themselves. If the effort needed to perform a task outweighs the benefit, users will regard the product as unusable.

3.2.1 The Scope of Usability

In the previous paragraphs we have referred to usability in terms of its relationship to "products" or as "systems" and the product's relationship to users. Both Human-Computer Interaction and usability refer to the interactions between users and products or systems. This is indeed convenient but when we speak of products or systems we are referring to a collective of various different components all of which make up the whole that is the software system. Such components include hardware, software, menus, icons, messages, user guides, quick reference guides, online help and training. All of these have a bearing on usability and conversely, the usability of each of these factors affects the usability of the system as a whole. This synergy between the components is echoed by Dumas & Redish (1999:6) who state that "changes in technology have blurred the lines among these product pieces, so that it is no longer useful to think of them as separate entities". As such, a user guide that is less than satisfactory from a user's point of view will adversely affect the overall usability of the system because user guides form a core part of the system. In the following section, we will examine users from the point of view of those cognitive processes and abilities which affect the way users use user guides.

3.3 THE HUMAN COGNITIVE SYSTEM

If we take a broad view, computers are information processing systems. Information, or data, is manipulated, created, modified, accessed and stored. Similarly, the human mind can also be regarded as an information processing system. As such, the broad model of a computer can be used as an analogy for describing the human information processor (Card *et al.* 1983:24; Downton 1991:20). We can draw several comparisons between the two contexts in that they can both be said to consist of memory, processors, interconnections, rules etc. However, such an approach can only be used for illustrative purposes as the structure of a computer does not necessarily reflect the structure of the brain. Indeed, there is still some debate about whether certain components of the mind are distinct physical locations or merely different functions of the same physical location (Card *et al.* 1983:23,36; Dix 1998:27; Faulkner 1998:33-34). Raskin (2000:12) warns against using current technology as figurative models because such models rapidly become outdated and quaint. Nevertheless, using computers as an illustrative model allows us to examine the human mind as a series of subsystems. If we return to the idea of a computer we can see on a very basic level that:

- information is input into the computer
- the information is processed, and
- an appropriate response or output is prepared

Applying this scheme to humans we can divide the human mind into the following subsystems (see Card *et al.* 1983:24; Downton 1991:20):

- the perceptual/sensory system
- the cognitive system
- the motor system

For our purposes here it is convenient to discuss the perceptual and motor systems together as they are similar to the basic notion of a computer's input/output system. This model, however, omits a fundamental factor common to both computers and humans: information is stored and accessed. And so, to make the model more accurate in terms of functions we need to incorporate memory into it. We can use the following components to examine the human system (see Dix 1998:12):

- input / output
- memory
- processing

3.3.1 The Human Input / Output System

As already stated, humans interact with the outside world and with computers through the exchange of information. This exchange relies on information being sent and received; in other words the input and output of information.

Information input for humans takes place through the five senses: sight, hearing, touch, taste and smell. For most people, the first three senses are the most important, especially in terms of HCI (Faulkner 1998:13; Dix 1998:13). Even though the senses of taste and smell are valuable senses for humans, it is not clear how they could be utilised in our interactions with software or documentation (Dix *ibid.*) and they will not be discussed further here. Similarly, the senses of hearing and touch, although invaluable for humans, are of little relevance in our examination of printed user guides; these senses will not be discussed here either.

3.3.1.1 Sight – The Visual Channel

The way in which humans see relies on a highly complex system which functions in two stages: the physical reception of visual stimuli and the processing of these stimuli (Dix 1998:14). While the physical limitations of the eye mean that we cannot see certain things (e.g. ultraviolet or infrared radiation etc.) the processing abilities of humans means that we can organise visual perception in terms of motion, size, shape, distance, relative position and texture (Downton 1991:14), even if some information is missing or incomplete (Dix 1998:14).

The Boundaries of Visual Processing

The basic principle underlying human sight is the reception of light reflected from the physical world. But this is only a small part of the visual channel. This light represents a stimulus which must then be interpreted. What makes this channel so valuable is what we do with the information we perceive. This information must be processed and transformed so that we can form an interpretation of the images we see.

In processing visual information, our expectations, experience and knowledge play a key role. For example, if we know that a truck is 15 feet high and 40 feet long, we will always perceive it as such even if we view it from a distance. It is this ability which allows us to make sense of unexpected, faulty, contradictory or incomplete information and allows us to resolve ambiguous information. Our expectations in relation to the world around us are largely determined by the context. Accordingly, one set of criteria may apply in one particular situation, e.g. a truck appears huge and has a trailer and 18 wheels when viewed up close, while different criteria apply at other times, e.g. a truck appears small and has a trailer and 18 wheels when viewed from a distance.

Unfortunately, this ability to make sense of ambiguous or contradictory information is not perfect and is prone to errors and interference. This can be illustrated by optical illusions such as the Müller-Lyer illusion shown on page 38. In the Ponzo illusion (Figure 6) the top line appears longer than the bottom line when in fact both are the same length. This can be attributed to an incorrect application of the law of constancy whereby the top line seems further away and is made to appear bigger (Dix 1998:19).

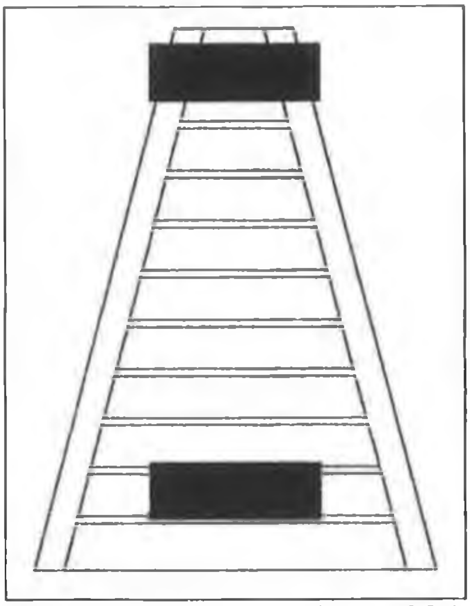


Figure 6: The Ponzo Illusion (Dix 1998:21)

the quick brown fox

jumps over the

the lazy dog

Figure 7: The Proof-reading Illusion (Dix 1998:21)

In the proof-reading illusion (Figure 7), most people reading the sentence quickly will miss the second "the". However, on closer examination, people will spot the mistake. This is an example of how our expectations compensate for unexpected information when reading.

Reading

Perhaps one of the most complex applications of visual processing, and the one most closely related to this study is the ability to read. Reading as an activity consists of several stages. Firstly, the visual pattern or appearance of the word is perceived. This physical image must be decoded by matching it against our own semiotic knowledge (signs such as letters and words) and then interpreted on the basis of syntactic and semantic analyses which operate on phrases or entire sentences (Dix 1998:22).

When reading, the eye moves in a saccadian manner. This means that the eyes do not move smoothly but rather in a stop-start manner. Each saccade consists of a brief period of motion followed by a fixation (Card *et al.* 1983:50) which is when the eye is at rest and when perception occurs. Fixations account for 94% of time spent actively reading. The eye moves both forwards and backwards to read and re-read text. These backward movements are known as regressions and are more frequent when reading complicated or difficult texts.

Generally speaking, the average adult can read approximately 250 words per minute. This speed means that it is unlikely that each letter is scanned and decoded in series. Indeed, according to Dix (1998:22) we can recognise certain familiar words by their shape just as quickly as we can recognise a single letter. One interesting effect of this is that it is very easy to destroy the visual clues which make whole words recognisable by shape. So for instance, if we were to capitalise a word, we undo the familiarity of the word's shape and consequently the word will have to be scanned and processed as a string of letters rather than as a single meaningful unit (*ibid.*). Take for example the word "intermediate". Written like this we can recognise it almost instantly. But if we write is in uppercase like this INTERMEDIATE, it is not so immediately recognisable.

3.3.1.2 Human Output

Taking a simplified view of the human cognitive system, we can say that information is received by the sensory organs and sent to the cognitive system for processing. Once the information has been processed, a response is produced. The brain sends the necessary impulses to the appropriate part(s) of the body in order to effect this response.

Our bodies can respond physically using our hands, fingers, thumbs, feet and voice. As with many functions and activities related to humans, the effectiveness and speed with which we respond physically varies from person to person as a result of factors such as age, fitness, health or alertness. The speed with which we react to a stimulus also depends on which sense receives the stimulus: we can react to an auditory stimulus in approximately 150ms; to a visual stimulus in 200ms and to pain in 700ms (Dix 1998:26). Reaction times are not the only factors affecting human output. The actual output rate varies depending on the part of the body used to respond to a stimulus. For instance, if we use one finger on each hand in an alternating manner, we can achieve somewhere in the region of 1000 key presses per minute. If we use just one finger, this figure is around 400 key presses per minute. If we use both our feet to respond, for example via pedals, we can achieve 600 presses per minute. With one foot this figure drops to 300 presses per minute. Vocal output allows us to achieve an output of between 180-400 words per minute (Downton 1991:26).

3.3.2 Perception

In the preceding paragraphs we examined the sense of sight. This is the most important sense in terms of how we use user guides and it provides us with information. Now we will look at what we do with the information we gather from our surroundings.

Perception is more that just seeing or hearing. Perception is a complex and active process which allows us to interpret information. By interpreting the raw information provided by our sensory organs we, in a sense, prepare it for further processing in the cognitive system. If it were not for perception, we would simply be receivers of sensory information but we would not be able to use this information for anything. Think of a motion detector – it can detect an intruder but unless it is connected to an alarm system it cannot activate a siren or alert anyone. Of course, if an alarm system had the cognitive processing abilities of humans, it would also be able to distinguish between intruders and friends. Conversely, without the sensor, the alarm system is deaf and blind – it simply cannot do anything because it receives no information.

3.3.2.1 Sensory Data Filters

With our powerful sensory systems, humans are under a constant barrage of sensory information. We receive enormous amounts of information through our eyes, ears, sense of touch etc. But it would be impossible for us to process all of this information or even be consciously aware of it all (Coe 1996:10). Indeed, we are only aware of a fraction of the sensory information we receive.

This is not a coincidence, for if we were to attempt to process everything we would waste valuable processing resources on things other than those we want to concentrate on. It is possible that such a volume of information could even overload our processing systems with less than desirable consequences. We must, therefore, organise and limit the sensory input so that we can process information in a structured and manageable way. This is done in a number of ways that make use of: *thresholds*, the *cocktail-party effect* and *sensory adaptation*.

THRESHOLDS

There are two types of threshold that we use to separate and organise sensory input: *absolute* and *just noticeable difference* (JND).

An *absolute threshold* is the smallest amount of stimulus we can detect 50% of the time. This type of threshold is largely dependent on the individual and each individual's psychological state. For instance, a user's motivation, expectations and experience are crucial in determining absolute thresholds when, for example, learning to use a new software application. Consequently, absolute thresholds are variable – the exact same stimulus may induce different responses under different circumstances and at different times (the time of day, whether the user is in a good mood etc.).

Just noticeable difference is the smallest difference noticeable between two stimuli 50% of the time. By way of example, let us consider a cup of coffee. Imagine we are gradually adding tiny amounts of sugar to the cup. We will not detect the presence of the sugar at first. If we continue adding tiny amounts of sugar, we will eventually begin to taste the sugar in the coffee. The difference between the point where we first detect the taste of sugar and the last point where we did not taste the sugar is the just noticeable difference.

COCKTAIL-PARTY EFFECT

The cocktail-party effect allows us to filter out information which is important or relevant and separate it from the deluge of sensory information we constantly receive. The effect allows us to focus in on important information while ignoring irrelevant information. The analogy comes from the notion of a cocktail-party where many different conversations are taking place. Amidst this bustle of information and conversation, we will hear someone mention our name over the noise in the room. Similarly, we can generally choose to listen to one particular conversation and effectively "fade out" the other conversations and "turn up" the conversation we want to listen to (Preece 1994:100; Coe 1996:12).

SENSORY ADAPTATION

Sensory adaptation describes the phenomenon whereby we become accustomed to a set of sensory inputs. For instance, if a person is working in an office and the air conditioning is turned on, the person may be distracted by the noise of the fan. However, after a while the person becomes accustomed to the new stimulus and no longer notices it. It is not until the fan is turned off and the noise stops that the person becomes aware of it again.

Thresholds, the cocktail-party effect and sensory adaptation are all mechanisms by which we select which information to process. They allow us to optimise processing resources and concentrate on what is actually important, relevant or of interest. (These mechanisms are also important factors in attention and cognitive processing which will be discussed in greater detail in Section 3.5.2). Now we have separated the information to be processed, we can look at how this information is interpreted and prepared for cognitive processing.

3.3.2.2 Ecological and Constructivist Approaches to Perception

There are a number of different theories which seek to explain how we turn basic sensory data into meaningful interpretations. These theories can be broadly categorised into the following groups: *ecological theories* and *constructivist theories.*

The fundamental difference between these two groups is that ecological theorists maintain that perception involves a process of gathering information from our environment to help us understand our surroundings. Constructivists, on the other hand, believe that visual perception is an active process based on what we actually see as well as our own previously acquired knowledge (Preece 1994:76). Using both of these elements we then construct an interpretation of the information we receive.

Ecological Approaches

This approach states that perception is a direct process whereby we detect information rather than create or interpret it. The ecological approach is not concerned with how we understand or recognise situations or scenes but rather what we need to know about a situation and how we go about finding it in our environment. This approach involves us actively exploring our surroundings and engaging in activities that allow us to find the necessary information.

Constructivist Approaches

The constructivist approach, on the other hand, maintains that visual perception is not just a direct representation of what we see but rather a model of our surroundings which is modified, transformed, enhanced and filtered using our knowledge, experience, expectations and memories. This approach sees perception as a process whereby what we see is compared against our experience of the world and an interpretation is constructed. What is more, by comparing what we detect from our surroundings against what we know, we can deal with a wide variety of situations and, if necessary, adapt to new situations and modify existing knowledge.

Piaget's concept of schemes (Piaget & Inhelder 1969:4; Hill 1995:15; Ginsburg & Opper 1988:20-22) is a useful tool in understanding this. When people are presented with new tasks or situations, they bring with them a set of existing ideas, methods and knowledge (known as a scheme) which they will use to tackle the task. However, if this scheme is not adequate for the task, they will modify this scheme in order to incorporate new knowledge, methods or approaches. Take, for example, the driver of a car. Driving a car requires a set of knowledge such as understanding gears, using the pedals, starting the engine, stopping distances, traffic regulations, manoeuvring the vehicle, etc. Now let us imagine that this person wants to drive an articulated truck. The knowledge of driving a car is only partly useful – the rules of the road still apply as does the knowledge of using gears. But the knowledge of manoeuvring, braking distances etc. is different for trucks and will have to be modified if the driver is to successfully drive the truck. Schemes are also referred to as perceptual sets (Coe 1996:16).

Grouping and Organising Information

In order to interpret the objects we see, we need to be able to regard them as meaningful units. Under the broad category of constructivist approaches, the Gestalt psychologists such as Koffka (1935) and Köhler (1947) developed a way of grouping or organising information so that it "means" something or forms something to which a meaning can be attributed. So rather than seeing a series of separate, individual objects in isolation, we group them into units or organised "wholes" (Coe 1996:18). This is the basis for the statement on page 60 that we can recognise certain words from their shape just as easily as from the individual letters that make up the word.

The Gestalt approach to organisation provides us with 6 basic "laws" which help us organise and interpret objects: *Proximity*, *Similarity*, *Continuity*, *Symmetry*, *Closure*, and *Common Fate*.

Proximity	If objects are near each other, the average person will tend to group them together. This law applies not only to objects such as lines or shapes but also to text, tables etc.
Similarity	If objects are similar, we will group them together as a unit.
Continuity	We are more likely to perceive smooth, continuous patterns rather than abrupt or non-continuous ones.
Symmetry	If we see an area bounded by symmetrical borders or objects, we tend to group everything together to form a symmetrical figure.
Closure	If we see an object with gaps in it, we will see past the gaps and view the object as a whole.
Common Fate	If we see objects moving or pointing in the same direction, we will group them together as a single unit because they share a "common fate".

Table 1: Gestaltist Laws for Grouping Information

PRÄGNANTZ

The law of Prägnantz (Coe 1996:23) is also called the "goodness of figures" and refers to the way humans generally opt for the simplest, most obvious interpretation of an object. This "law" illustrates how we group information and compensate for missing or faulty information to produce the most probable and likely interpretation given the context. In a way which is similar to the ideas put forward by supporters of relevance theory (Sperber & Wilson 1986; Gutt 1991), humans will opt for the interpretation which is most accessible and which requires the least processing effort.

Pattern Matching

Once we have grouped the objects we see into meaningful units, we need to recognise them in order to understand what they are. There are a number of ways in which we can recognise shapes (or patterns) and which ultimately determine whether we correctly interpret them.

Prototype Matching

This method involves us storing a general, fundamental shape or pattern against which we compare objects to find a match. Essentially, this model is a very basic stylisation which is fundamentally the same regardless of any cosmetic or superficial differences we encounter from instance to instance.

Template Matching

In contrast to prototype matching which provides us with a general outline of objects, template matching involves us storing detailed patterns of each and every variation of an object we see. So rather than having a prototype for the letter "P" which states that a "P" consists of an upright line with a loop attached to the top right, template matching means we need a model or design for each "P" we encounter.

Distinctive Features

This method involves us distinguishing objects on the basis of their distinctive feature patterns. For example, a car has four wheels while a bicycle has just two wheels. With this method, we recognise objects by analysing them and matching distinctive parts of an object as opposed to the entire object.

3.4 MEMORY

Having discussed the sensory system we will now continue our examination of the "infrastructure" which allows the human cognitive system to work. From a human-computer interaction (HCI) and learning point of view, we can say that the sensory system is the mechanism for receiving data to be processed while memory is the mechanism which facilitates cognition and learning. Only by understanding memory can we proceed to look at how data is processed and understand how we learn and solve problems.

Memory is fundamental to virtually every one of our actions from reading, eating and walking to writing, learning and speaking. Without it we would not know what to do with the information we receive through our senses. At its most basic physiological level, memory is "a physical change in the neuronal structure of the brain" (Coe 1996:69). When information is added to our memory it creates new neuronal pathways and connections.

There are three types of memory:

1. Sensory Memory
2. Short-term Memory (STM)
3. Long-term Memory (LTM)

These three types of memory work together, passing information between them to allow us to carry out cognitive processing.

3.4.1 Sensory Memory

Sensory memory, also known as *sensory registers* (Coe 1996:72) or *sensory buffers* (Dix 1998:27), is the first stage of memory. It is an area of conscious memory which acts as a buffer, temporarily storing information received through the senses. Each of our senses has its own sensory memory (Coe 1996:71), e.g. *iconic memory* for visual stimuli, *echoic memory* for aural stimuli and *haptic memory* for touch (Dix 1998:27). This type of memory acts as a temporary storage area for sensory information before it is passed on for processing. Information stored here is unprocessed, i.e. it remains in its physical form and is not decoded (Downton 1991:22). In effect, this means that the information stored here is extremely detailed and accurate. However, because of the limited capacity of sensory memory, information stored here is the most short-lived and is constantly being overwritten. In general, information is stored in sensory memory for anything between 0.2 seconds (Downton 1991:22) and 0.5 seconds (Dix 1998:27) although echoic memory is more durable and lasts for approximately 2 seconds (Downton *ibid.*).

The existence of iconic memory can be demonstrated easily using the concept of persistence of vision – the principle upon which television and cinema work. By displaying a series of separate images in rapid succession, the eye is "tricked" into seeing a single moving image. Similarly, echoic memory can be illustrated by those instances where we are asked a question and we ask for the question to be repeated only to discover that we actually heard it after all. In a manner of speaking, sensory memory allows us to replay information and gives us a second chance to process information. Sensory memory also serves as a route to short-term memory (STM) for the sensory information we receive (Dix 1998:27; Coe 1996:71). However, due to the brief duration of sensory memory, not all perceptions become proper memories (Raskin 2000:18).

3.4.1.1 The Low-Capacity Channel

Linking sensory memory to STM is what is termed the "low-capacity channel" (Downton 1991:23). This channel serves as a conduit for information passing from sensory memory to STM. In practice, however, this channel has a low transfer capacity, something which is evident from the difficulty we experience in paying attention to many different sources of information simultaneously. In addition to transmitting information, this channel also converts the information from its raw, physical and unprocessed state into symbolic representations which can be used in STM. Indeed, this is where perception as described in Section 3.3.2 occurs. The limited speed with which this information is converted helps to explain the low capacity of the channel in general. This limitation means that the channel is very prone to overloading.

3.4.2 Short-Term Memory (STM)

A popular way of explaining the concept of STM is to describe it as a "scratchpad" or as RAM in a computer (Dix 1998:28; Hill 1995:19). STM is responsible for storing information that we are currently using. It is where we carry out all of our memory processing, encoding and data retrieval. STM allows us to "do" things with information. We can also filter information here and discard information which is no longer needed.

Card *et al.* (1983:38) argue that STM (or working memory as they call it) is really only an activated subset of information stored in long-term memory (LTM). While it is true that STM obtains some of its input from LTM, e.g. stored knowledge, procedures etc., information passed on from sensory memory also provides STM with its input.

In contrast to information stored in sensory memory, information in STM is stored in the form of symbolic representations or schemes (Coe 1996:72). However, like sensory memory, information is stored in STM temporarily. The information is lost, overwritten or replaced after 20-30 seconds (Downton 1991:24), although with practice information can be retained for several hours (Coe *ibid.*). That information is only stored temporarily is due to the limited capacity of STM. In 1956 Miller posited that the capacity of STM is seven chunks plus or minus 2 chunks. This "7±2" rule is universally accepted as fact (Faulkner 1998:34; Coe 1996:72; Downton 1991:23; Dix 1998:28) and is generally true for most people. This can be illustrated using the following sequence of numbers:

- 0352765994

The average person may find it difficult to remember each digit in this sequence. However, if we group the digits into smaller sequences as we would with a telephone number, each sequence can be treated as a single chunk:

- 035-276-5994

So, instead of remembering ten separate pieces of information, by chunking the information we reduce the amount of space required to remember them. An interesting property of chunks is that what actually constitutes a chunk depends on individual people and the content of their LTM (Card *et al.* 1983:36). According to Downton (1991:24) the number of chunks which can be stored is independent of the amount of information each chunk contains. We can, therefore, combine small chunks to form larger chunks and so on. For example, letters (small chunks) form words (larger chunks) which can be combined to form sentences (even larger chunks) and so on (Faulkner 1998:73). With sufficient practice and rehearsal in STM, several sentences can be treated as one single chunk.

3.4.3 Long-Term Memory (LTM)

Long-term memory is the final part of our memory system and it is here that information is stored once passed on from STM. Whereas capacity and retention are key factors when discussing sensory memory and STM, they do not apply to LTM as this type of memory is essentially unlimited in its capacity and information is stored there forever (Faulkner 1998:35; Coe 1996:74; Downton 1991:25; Dix 1998:30).

It is widely held that there is really no such thing as "forgetting" information, rather the information is still stored in LTM but that as the memory grows older, the traces which lead to the information and help us locate it become increasingly faint and less defined. The result is that we simply cannot find the information we want (Faulkner 1998:35). Over time, it becomes more difficult to locate information and this can lead to this information being permanently "forgotten". This condition is exacerbated by the fact that information which is most recently and frequently used is easiest to retrieve from memory (Downton 1991:25).

Of greater interest when discussing LTM, however, is its structure and how information is retrieved from it. LTM can be divided into two types: *declarative memory* and *procedural memory*.

3.4.3.1 Types of Long-Term Memory

Coe (1996:74) divides LTM into two primary categories - *declarative memory* and *procedural memory*. Declarative memory is described as "memory what" (*ibid.*). This is memory of events, facts and images. "Memory how" or procedural memory is memory for motor skills, cognitive skills, reflexes, how to do things etc.

Declarative Memory

Declarative memory consists of a number of different types of memory: episodic, associative, lexical, image memory and semantic.

Episodic memory is our memory of events and facts relating to them (Dix 1998:31; Faulkner 1998:37). This memory allows us to reconstruct certain events and remember facts about them. Coe (1996:75) also mentions a specific type of episodic memory which is like a high-resolution snapshot of a particular event that was particularly surprising, emotional or otherwise significant. This is known as *flashbulb memory* and is attributed to momentous occasions be they good or bad.

Associative memory is the way we remember information using tags with which we label schemes of knowledge (see also page 64).

Lexical memory is what we use to remember the graphical and phonological features of words. This refers strictly to the physical properties of words – the combination of black and white lines on paper, for example. The **meaning** of these words, however, is stored in *semantic memory*.

The term *image memory* can be quite misleading because it does not only refer to physical pictures we have seen and stored in memory but also to mental images which we construct on the basis of events, pictures, situations, words etc. For instance, we can picture in our minds a childhood birthday party or a beautiful place we may have seen on holiday. We can also manufacture mental images without ever having seen the physical object or place in question. This is part of what we refer to as *imagination* and it is a product of our image memory. For instance, we can picture ourselves sipping frozen margaritas on the deck of a yacht in the Caribbean even though we may have never been on a boat in our lives, nor visited the Caribbean.

This type of memory is more durable and reliable than any other type of memory (Coe 1996:77). When we store information, either in the form of an image or accompanied by an image, we can recall it more readily than information that does not have image associations. For instance, it is easier to remember where the dipstick is on a car if we associate a visual image of its location rather than a verbal description alone, e.g. a long piece of metal protruding from the oil sump in a car's engine.

Semantic memory is our memory for facts, meanings, concepts, vocabulary etc. Semantic memory is our knowledge about the world – a structured record of facts, knowledge, concepts and skills that we have acquired. Semantic memory is structured to allow orderly and reliable access to the information (Dix 1998:31) One model used to explain how this information is structured is that of the network. Classes are used to relate items together and each item may inherit attributes from superordinate classes (Dix *ibid.*). The classes are all linked together in a network. Semantic networks, however, do not allow us to model the representation of complex objects or events. They merely allow us to model relationships between single items in memory (*ibid.*).

Another structure proposed to counteract this problem is the notion of *frames* and *scripts*. This model organises information into data structures. These structures contain *slots* which contain default, fixed or variable attributes. Scripts are an attempt to model the representation of stereotypical knowledge about given situations (Dix 1998:33). This representation allows us to interpret partial descriptions fully. The frames and scripts are then linked together in networks to present hierarchically structured knowledge.

Another type of knowledge representation is the representation of procedural knowledge. This is our knowledge about how to do something. Procedural knowledge is generally represented in a production system consisting of "*if-then*" condition-action rules (Dix 1998:34). With this model, information that is received in STM is matched against one or more of these rules and the associated *action* is determined by the *then* part of the "*if-then*" rule. For example:

If the traffic light is green	*Then* drive through the junction
If the traffic light is red	*Then* stop and wait until it turns green

The following section describes procedural memory in more detail.

Procedural Memory

Procedural memory is acquired using a number of processes: motor skill learning, cognitive and perceptual learning as well as classical conditioning, priming, habituation and sensitisation.

Motor skill learning is the means by which we remember how to do physical activities like blinking, moving our fingers, pushing buttons etc.

Perceptual learning is the process of learning **how** to perceive sensory information each time we encounter it. For instance, the first time we try to ride a bicycle, our senses will tell us that we are not balanced and that we are falling. As a result we may pull our arms up to protect our head as we fall to the ground. With practice, however, this sensory information results in us making slight changes in our body position to correct the loss of balance and continue cycling.

Clinical or Pavlovian conditioning (Brainviews 2002) is our memory for a response that is caused by a stimulus and a reinforcer. Drawing its name in part from experiments conducted by Ivan Petrovich Pavlov (Fredholm 2001), this type of memory continues even without the reinforcer.

Priming is the process whereby triggers or cues which activate information from memory are stored. Priming memory is short-lived and available only through the sense that activates it – it cannot be activated or accessed by any other sense. Furthermore, this type of memory does not include the subsidiary information such as when and where the memory occurred.

To put this in context, let us take the following example: if you were asked to think of the colour red and were then asked to think of a particular type of flower and then an emotion, you might think of a rose and anger. The word "red" acts as a trigger which is temporarily stored and which activates other information from memory.

Habituation is the process whereby we become accustomed to sensations repeated over time. If we think back to the example of office air conditioning on page 63, we can see that habituation is the memory that allows us to decrease our attention to the noise of the fan after a certain amount of time.

Sensitisation is the process whereby we acquire sensitivities to specific events, situations or actions. If, for example, you were bitten by a dog as a child, the mere sound of a dog barking may provoke an extreme reaction as an adult.

3.4.4 Retrieving Information from Memory

There are two ways of retrieving information from memory: *recall* and *recognition*. In the case of recall, we reproduce the information from memory whereas with recognition, the information presented informs us that we have seen the information before. Recognition is less complex than recall because the information is provided as a cue (Dix 1998:36).

However, because recall actually reproduces the information and not just the knowledge that we have seen it before, it makes sense to try to assist the recall process. When we try to recall information from LTM, we do not know what the information is or what the cues that aid retrieval are (Card *et al.* 1983:82). However, if we place cues in STM we can assist recall. The problem here is that if we add cues to STM we rapidly fill the STM capacity of 7±2 chunks of information. The result is that while we speed up retrieval, we actually slow down processing in STM. In the case of user guides, the text itself can be used to place cues in STM, but if too many cues are added, the reader's progress through the text and task will be slowed down.

Given the fact that we can recognise information far more easily than we can recall it (Preece 1994:118) it is useful to have users recognise the information they need to perform a task rather than recall it. Of course, there is a trade-off between recall and recognition. Whereas recognised information is easily retrievable whenever the information is present and it does not require learning, recall can be much more efficient because information does not need to be located and assimilated. If, however, information is repeated several times, it will in time become automated in procedural memory, and subject to recall rather than just recognition (Dix 1998:34; Raskin 2000:18-20).

3.5 Cognitive Processing

So far we have examined ways in which we can describe the human cognitive system and we have looked at the components of this system as well as some of their capabilities and limitations. We will now examine how these subsystems interact with each other and function together as a whole to make the human cognitive system what it is – an information processor.

This section looks at how and where we use this system, i.e. what we use our cognitive system to process as well as the actual mechanics involved in using the system. For our purposes, this discussion will be restricted to how we tackle new information and tasks and how we learn.

3.5.1 Cognitive Conscious and Cognitive Unconscious

In his discussion of human cognitive processes, Raskin (2000:11ff) distinguishes between the *cognitive conscious* and *cognitive unconscious*. This distinction is necessary in order to explain the way in which humans actually go about processing information and, perhaps, to shed light on the limitations and anomalies of how we perform tasks.

Human cognitive unconscious essentially refers to information which we are not consciously using or aware of at a given point in time. We can refer to the cognitive unconscious as those things we are only subconsciously aware of but which are not relevant to what we are currently doing (compare this with the notion of relevance as espoused by Sperber & Wilson 1986 and by Gutt 1991). Conversely, our cognitive conscious refers to information, tasks, etc. that we **are** conscious of, i.e., that we are currently using.

Another way of looking at the difference between cognitive conscious and cognitive unconscious is that when we access and process information we are transferring it from our unconscious to our conscious. This transfer of information can be triggered by a stimulus, such as reading a sentence, or by an act of volition. For the purposes of this study, we can say that cognitive conscious broadly correlates to our everyday notion of attention.

3.5.2 Attention

As was discussed in Section 3.3.2.1 our sensory system is under a constant barrage of information and input. We mentioned that in order to function effectively and avoid sensory information "overload" it is essential that we be able to filter and group information in order to extract and absorb what is immediately of relevance to us. But why is this necessary? The notion of avoiding overload is true to a certain extent but the underlying principle is that of attention, or to quote Preece (1994:100), selective attention.

Coe (1996:9) describes attention, or rather attentive processing, as processes that involve cognitive functions such as learning, memory and understanding. *Attentive* processes involve higher cognitive functions. This is in contrast to *preattentive* processes which do not involve cognitive processing and which are primarily a function of sensory input (*ibid.*). So we can see that attention is similar to, if not the same as, Raskin's (see Section 3.5.1 above) concept of cognitive consciousness or information that is currently being processed in STM.

3.5.2.1 *The Locus of Attention*

We have a certain degree of control over which information we process in STM. In other words, we can, to a certain extent, control which information is the subject of our attention. For instance, we can be driving home and performing all of the processing necessary in order to do this task and we can then start thinking about what we would like for dinner. In this way we can make unconscious information conscious.

Raskin (2000:17) urges caution with regard to using the word "focus" in relation to attention primarily because *focus* can be used as a verb, and as such it implies some aspect of volition or choice on our part. This, he maintains, can lead to misunderstandings as to the true nature of attention. Instead he uses the expression *locus of attention* to refer to the current object of our attention regardless of how it came to be such. In other words, the locus of attention refers to information that is currently being processed in STM. He justifies this shift in terminology from the widely used "focus" on the basis that while we can deliberately shift our attention to another task, issue or subject (e.g. the cocktail party effect as described on page 62), our attention can be focussed for us by means of other stimuli, e.g. people, events or objects. However, in keeping with common usage in the relevant literature, we will continue to use focus as a verb in the following paragraphs. The preceding caveat regarding volition should, however, be remembered.

This lack of complete control is evident from the following examples: if you are told not to think of ice-cream, the likelihood is that you will think of it. Similarly, if you are thinking about ice-cream, you cannot make yourself stop thinking about it and make the information unconscious unless, of course, you shift your locus of attention to something else.

A key feature of the locus of attention is that we only have one (Raskin 2000:24).We have often heard people say that they can do only one thing at a time, particularly when they are busy. Apart from the obvious physiological constraints preventing us from performing tasks (e.g. we cannot make a cup of tea while at the same time vacuuming the carpet) the reason for this is because in general terms we can focus on and process only one thing at a time. This fact is explained by Card *et al.* (1983:42) who explain that the cognitive system is parallel in terms of recognition but serial in terms of its actions or responses. This means that we can be aware of several things at the same time but we can only do one deliberate thing at a time. An example of this would be trying to hold an in-depth conversation while listening to the radio. So as a general rule, our attention is focussed on a particular thing (Preece 1994:101). This claim may seem less than credible or quite simply impossible. After all, what about people who can continue a conversation while they are driving? The answer is simple, although perhaps not obvious.

3.5.2.2 Attention & Selection

When discussing the fact that attention can be either voluntary or involuntary, Preece (1994:101) refers to competing stimuli "grabbing our attention". Herein lies the explanation for the apparent existence of multiple loci of attention. Instead of being able to focus on multiple tasks, our locus of attention switches from one task to another. Raskin (2000:24) also acknowledges this point. When describing how events can trigger conscious attention, he stresses the point that we have not gained an additional locus of attention but rather our locus of attention has been shifted elsewhere (see also Card *et al.* 1983:42). Preece (1994:105) later refers to multitasking which is ostensibly the same as what she calls divided attention or multiple loci of attention. Multitasking is, in fact, "continually switching between different activities rather than performing and completing tasks in a serial manner". Both Preece (*ibid.*) and Raskin (2000:16) acknowledge that our ability to perform tasks is sequential or serial rather than truly parallel.

But the question arises here of how we switch our attention from one task to another. After all, if we are focussed on one task, how can we switch to another task if we are capable of consciously processing and responding to only one task or stimulus? This a wide-ranging and problematic question in cognitive psychology and it would be impractical to discuss every aspect of this issue here. Instead, we will discuss the main approaches to how attention switches from one task or stimulus to another.

But before we embark on this discussion of attention, however, it would serve us well to quickly recap on preceding paragraphs. We know that our senses are constantly receiving information and that this information is stored temporarily in sensory memory or registers. We also know that attention fundamentally refers to the active, conscious processing of information at a given time. This means that just one of the numerous sources of information or stimuli is being processed. Attention is, therefore, the process by which we select information from the mass of incoming information and process it.

Numerous theories have been formulated over the decades to account for the way our attention changes focus to concentrate on various stimuli. Fundamental to all of these theories is the question of what happens to the "unattended" information

(Ellis & Hunt 1993:50), or rather the information we are not paying attention to at any given moment. The main approaches to answering this question are grouped under Bottleneck Theories (Gavin 1998:34) and Capacity Models (Gavin 1998:37; Ellis & Hunt 1993:50-52)below.

Bottleneck Theories

Bottleneck theories fall under the categories of early selection and late selection models and they generally revolve around some variation on the notion of filters. Indeed, the idea of filters is a key element of both early and late selection theories. If we cast our mind back to the idea of the cocktail party effect on page 62 we will recall that we can filter out stimuli and focus on one particular stimulus.

Early Selection Filter Models

In early selection filter models, we work on the assumption that only one source of information can be processed (Ellis & Hunt 1993:52). Logically, this means that unattended information is filtered out before cognitive processing takes place, i.e. before the information reaches STM. We can see, therefore, that early selection takes place early on in the information-processing chain of events.

Perhaps the most well known early selection filter model is *Broadbent's Switch* (Ellis & Hunt 1993:52ff). Broadbent (1958) proposed that our attention is determined by a filter and a detector located between sensory memory and STM (Gavin 1998:35). Using the idea of a switch means that we process information from one source or channel only in an "all or nothing" manner (Ellis & Hunt *ibid.*). Essentially, if one stimulus is being processed, all other stimuli are effectively blocked out. But if we are blocking all of the remaining sources of information, how do we remain sufficiently aware of our surroundings to be able to operate the switch and shift our attention? How do we decide how, when and where to focus our attention? According to Broadbent, the unattended information is subjected to a pre-attentive analysis (i.e. it is analysed before we become aware or conscious of it) which examines the physical nature of the incoming information. From our discussion of sensory memory on page 68 we should recall that information is stored here in a detailed and unprocessed form. This means that any other form of analysis of the information would be impossible before the information is passed on to STM. Information which is selected on the basis of physical characteristics is then passed along the low capacity channel and into STM for processing. The remaining information is left in sensory memory where it decays and disappears after 0.2-0.5 seconds.

Unfortunately, the notion of an "all or nothing" switch does not explain the cocktail party effect. If we are concentrating on one particular conversation to the exclusion of all other sensory input, how can we detect our name being spoken and change the focus of our attention? The audio input arising from our name being spoken is not processed cognitively and as such, the physical representation of the sound in sensory memory has no meaning for us. This problem is also highlighted by Gavin (1998:36) and Ellis & Hunt (1993:54-56) in their discussions of experiments carried out by Cherry (1953) and Treisman (1960). Treisman discovered during experiments involving dichotic listening and shadowing[1] that subjects were able to report back on some of the content of the unattended information and that the unattended information could even affect the performance of the attended, shadowing task (Ellis & Hunt 1993:55). It is obvious from this that the unattended information is subject to at least some limited form of cognitive processing. This presents obvious problems for the application of the basic switch model.

The Attenuator Model

In light of the problems associated with the switch model highlighted in various experiments, Treisman (1964) developed a more flexible theory for the early selection of information. Rather than using a simple, binary switch, Treisman proposed the use of an attenuator. An attenuator is a type of switch which is used to control the volume or speaker balance on stereo or radio equipment. Instead of a simple on / off function, an attenuator controls signals in stages or gradations to allow more or less signal through.

To give a practical example of this, if we think of the speaker balance on a stereo we know that if we turn the dial (attenuator) to the left, more sound passes through the left speaker channel and less passes through the right speaker channel. If we turn the dial to the right, the opposite happens. So rather than all of one channel and none of the other channel passing through for processing, we have a situation

[1] Dichotic listening involves subjects wearing headphones and simultaneously listening to two separate audio inputs or speech recordings, the first through one earpiece and the second through the other earpiece. Shadowing involves subjects repeating verbatim the speech that is heard through one of the earpieces.

where virtually all of one channel and **some** of another channel are processed. Like Broadbent's switch, all input is physically analysed and filtered before it reaches STM (Ellis & Hunt 1993:57). Unlike Broadbent's switch, however, all of the information passing through the attenuator can conceivably pass through to semantic processing (Gavin 1998:36). The difference here is that some of the information is "turned down". Gavin (*ibid.*) maintains that such a model does not make "cognitive sense" and does not make effective use of the cognitive system because the level of processing carried out on unattended information is not far from full cognitive processing. Such a situation would undoubtedly use up all of the cognitive system's resources. Ellis & Hunt (1993:58) state that the attenuator model is frequently regarded as being too cumbersome to be practical.

Late Selection Filter Models

In contrast to the early selection models outlined above, late selection models operate after information has undergone partial cognitive processing. Information stored in sensory memory is passed to STM where pattern matching occurs (see page 65). When the information activates its LTM representation, i.e. a match is found for the pattern, a response is initiated. However, the human cognitive system is only capable of handling one response at a time (Ellis & Hunt 1993:58; Card *et al.* 1983:42). This means that instead of being able to focus our attention on one *input* only, we are, in fact, only able to concentrate on one *output*, namely the item of information activated in LTM.

Late selection functions on the basis of all information being passed in parallel into STM. There are obvious limitations as a result of the low capacity channel linking sensory memory and STM (see page 68) but for the most part, all information is passed through. Given the fact that STM is limited in its capacity, only some of the information can be stored there. The decision as to which information is stored in STM is based on the relative importance of the information. Information relating to the task at hand is assigned greater importance (Gavin 1998:37). Information which is deemed to be important is then subjected to more rigorous processing. While several items of information may be processed, we can organise and handle responses to one of them only.

Preconscious Processing and Semantic Priming

A crucial characteristic of the late selection model is that all information is processed to some extent. As attention selection takes place after recognition of information and at the point where responses are selected, the unattended information can sometimes be put to good use. Although we can only deal with one response at a time, the fact that we partially process so much information has an effect on the content of STM and which parts of LTM are activated.

Semantic priming is a phenomenon whereby in cases where "the stimulus which preceded the current stimulus was semantically related to the current stimulus, the response to the current stimulus is affected by the preceding stimulus" (Ellis & Hunt 1993:60). This means that stimuli can pave the way for subsequent related stimuli. In effect, semantic priming activates or facilitates access to related items of information. This makes the recognition of subsequent information easier (*ibid.*).

Capacity Models

Capacity models work on the assumption that the human cognitive system has a limited, finite set of resources. When we are aware of more than one stimulus, the available resources are allocated in varying amounts to different stimuli. For instance, a complex stimulus will require more of the available resources with the result that there are fewer resources available for processing other stimuli. Accordingly, attention can thus be defined as "the process of allocating processing resources or capacity to various inputs" (Ellis & Hunt 1993:62). This model is demonstrated in what Ellis & Hunt (1993:63) refer to as the *secondary technique and cognitive effort*. This technique is used to show how much cognitive effort (or processing resources) is needed for a secondary task while a person is already performing a primary task. Experiments conducted on this basis showed that the more complex the primary task, the more difficult it is to perform a second task. Interestingly, the greater the cognitive effort required for a task, the better the memory for that task and related information (*ibid.*).

Gavin (1998:38-39) characterises the model as follows: Competing stimuli produce non-specific interference – the two tasks do not directly interfere with each other but they do compete for the same finite set of resources. The more resources that are needed for one task, the fewer resources are available for the other task. It follows, therefore, that we can theoretically perform two tasks provided the total processing demand does not exceed the available capacity. When total processing demand exceeds the available capacity, performance on one task deteriorates (cf. Raskin 2000:21). The allocation of resources is flexible and can be adjusted to meet the demands of stimuli.

Performing Simultaneous Tasks

So what happens when we try to perform competing tasks? What happens when our attention is divided? Preece (1994:105) distinguishes between primary and secondary tasks. Primary tasks are those tasks which are most important at that particular time. In order to successfully multitask, Preece maintains that we need to be able to switch rapidly between the various tasks. The task currently being carried out is said to be *foregrounded* while the other tasks are temporarily *suspended*.

In principle at least, this appears to be a more than adequate solution to the problem of multitasking. There are, however, a number of problems associated with multitasking and which detract from the apparent benefits of performing multiple tasks at the same time. The first problem is that people are very prone to distraction (*ibid.*). When switching between tasks, our attention is temporarily not focussed, leaving our attention prone to becoming focussed by other stimuli. Also, there is a tendency to forget where we left off when we return to a task with the result that we can return to a task at a different point from where we left it. Another problem associated with switching tasks is that the more intensely we are concentrating on a task, the more difficult it will be to switch our locus of attention to a new task (Raskin 2000:20).

Raskin (2000:21) also states that when we attempt to perform two tasks simultaneously, our performance on each task degrades. This phenomenon is known as interference and can be explained by the fact that both tasks are essentially competing for the same finite amount of processing capacity (i.e. the "7±2 chunks" rule on page 69). An important point to bear in mind here is that Raskin is referring to two tasks which are not automatic and as such require large amounts of processing capacity. But what are automatic tasks and how do tasks become automatic? To understand this we will need to examine the cognitive processes involved in learning.

3.5.3 Learning

As we saw in preceding sections, long-term memory is a vast and virtually unlimited mechanism for storing information. Just as not all perceptions become memories (see page 68), not all information is stored in memory; it must first be learned. What is more, we cannot "force" information to be stored in long-term memory. Learning can be defined as a relatively permanent change in behaviour as a result of experience. Novak (1998:19ff) distinguishes between two principal types of learning: *rote learning* and *meaningful learning*. Rote learning is, according to Novak, not true learning and is not ideal because it is an arbitrary, verbatim and non-substantive incorporation of new knowledge into a person's cognitive structure. Rather than learning new information and integrating it with existing concepts and relating it to existing experiences, rote learning involves memorising information in a "word-for-word" fashion. And because the information is not related to existing knowledge, it is very difficult if not impossible to become proficient enough to use the information independently and creatively. All that is remembered is the literal message and not the knowledge or meaning behind it (Novak 1998:20).

Meaningful learning, on the other hand, is a "process in which new information is related to an existing relevant aspect of an individual's knowledge structure" (Novak 1998:51). This type of learning is the best type of learning because it is a non-arbitrary, non-verbatim and substantive incorporation of new knowledge into a person's cognitive structure. This means that information learned is linked with existing knowledge and concepts and with higher order concepts. Essentially, the learner understands the information, how it relates to existing knowledge and ultimately the learner understands how to use the newly acquired information. Meaningful learning, however, requires a conscious decision on the part of the learner in order to take place (*ibid.*). It cannot be forced, merely facilitated.

3.5.3.1 The Basis of Knowledge

Although it may seem obvious, when we learn we are acquiring new knowledge or, more specifically, meanings. But what is meaning? Novak presents the terms *concept* and *proposition* as the basic units of knowledge and meaning. Concept is defined as "a perceived regularity in events, objects or records of events or objects designated by a label" (Novak 1998:22). A concept is, therefore, a name given to a series of things or events that we observe to have certain similarities. Propositions, on the other hand, consist of two or more concepts which are combined to form a statement about an event, object or idea. They are the principal units of meaning. Propositions can be valid, invalid or nonsensical. Novak (1998:40) likens propositions to molecules while concepts are compared to atoms.

3.5.3.2 The Components of Meaningful Learning

There are, naturally, several stages involved in meaningful learning and it would be impractical to discuss all of them here. However, we will discuss three of the key components of learning here: *concept learning, representational learning* and *propositional learning.*

CONCEPT LEARNING

If concepts are the smallest meaningful units of knowledge, it holds that they are the first thing that will need to be acquired in order to learn new information. This process is called concept learning and it is the subject of much debate. There are those such as Piaget who maintain that in order to acquire a new concept, we must first perceive the regularity to which the concept refers before we can acquire the label. This may or may not be true. Theorists such as Vygotsky (1962) believe that having a label already stored in memory can actually assist in the acquisition of the concept.

REPRESENTATIONAL LEARNING

Representational learning is a type of meaningful learning where we learn a word, sign or symbol that acts as a label for an object or event. An example of this is the way we learn proper nouns. As mentioned earlier, we can either learn labels before we learn the concepts or we can learn concepts before we learn the labels. Similarly, representational learning may take place either before or after concept learning. However, representational learning on its own is insufficient in terms of meaningful learning because the concept is not acquired and there is no meaning or interrelationship with other knowledge.

PROPOSITIONAL LEARNING

If concepts are like atoms and propositions are similar to molecules, it follows that out of a small number of concepts it is possible to create a large number of combinations (or molecules). In practical terms, the meaning we acquire for a concept is formed from the composite of all the propositions we know that contain the concept. The more propositions we have that contain the concept in question, the richer our understanding of the concept will be (Novak 1998:40). And so, propositional learning is the process of linking, integrating and associating concepts to provide richer and more detailed meaning. The processes by which we acquire and combine new concepts and propositions are described below.

Acquiring New Information

There are two primary ways in which we acquire new knowledge: *concept formation* and *concept assimilation.* Concept formation involves constructing meanings for words from observed regularities. To illustrate this, imagine we have seen lions, tigers, cats and dogs and they all eat meat. When we learn that this common activity makes them carnivores, we form the concept of *carnivore.*

With concept assimilation, we acquire meanings for concepts by associating them into propositions which contain already known concepts. This can be illustrated using the example of *scone.* We know *scones* are a type of *bread* which in turn is a type of *food.* Here the new concept – scone– is subsumed beneath the concept of bread which is in turn subsumed beneath the concept of food. In this example, food and bread are subsuming concepts. The process of subsumption results in changes not only to the new concept but also to the concepts which subsume it. Consequently, information recalled about scones may be slightly different from that which was originally learned. Similarly, if over the passage of time, the concept of scone is forgotten or cannot be described adequately – a process known as obliterative subsumption – it will have modified associated information sufficiently to provide enhanced information about that particular concept area. So while we may not remember the precise details of the information we learned, we will still recall the knowledge it produced as a result of being learned. The process of concept assimilation is never fully finished because we are continually adding or associating new concepts with existing ones (Novak 1998:59-61).

3.5.3.3 Learning Processes

In general there are two main approaches to learning theory: behaviourist and cognitive. Behaviourist learning theories focus on objective, quantifiable behaviour rather than on mental acts which we cannot observe. They are concerned with the connection between actions, the role of reward in behaviour etc. Cognitive learning theories focus on mental acts such as conceiving, believing, expecting etc.

According to Coe (1996:34) we learn using a combination of behaviourist and cognitive approaches. The components of learning include:

- experience,
- schemes,
- habits,
- reinforcement,
- interference,

Experience & Schemes

We learn from experience. When we are met with an experience or situation we either create a new scheme or we use/modify an existing one. Thus any information provided, for example, in an instructive text such as a user guide, must either take advantage of readers' existing schemes or help them create new schemes quickly and easily. The easiest way to leverage existing schemes is to give examples based on schemes they already have.

Habits

Habits are learned connections between a stimulus and a response. The strength of the connection is called the habit strength. Related habits are grouped into habit families, each of which has a hierarchical pecking order. The most effective habits which we tend to use first or most frequently are located higher up in the hierarchy.

New habits can be introduced by comparing and contrasting old habits with new habits or building on existing habits. We will discuss habits in more detail below.

Reinforcement

Reinforcement is the process of using events or behaviours to produce learning. These are known as reinforcers and they can be either positive or negative. If we relate this idea to software user guides, one possible example of positive reinforcement would be if it tells a new user about using the keyboard shortcut CTRL-P to print a document. Each time the user does this, the document is printed, thereby reinforcing the knowledge that the shortcut works and is another way of printing a document.

Negative reinforcement involves the removal of an unpleasant or undesirable situation. For instance, if a user accidentally deletes all of the text in a document, pressing CTRL-Z will undo the previous action and restore the deleted text. This removes the undesirable condition and reinforces the user's knowledge of the undo function. In this way, the information in a user guide reinforces learning of functions by allowing users to do something useful or to correct problems.

As well as the positive/negative dichotomy, reinforcement can be divided into the following types:

Continuous

Continuous reinforcement takes place each time an event occurs. This is the quickest way of promoting learning and it establishes very strong expectations for the user that reinforcement will always take place. The result of this is a dependence on the part of the user and consequently if the reinforcement stops, the learning stops too.

Intermittent

Intermittent reinforcement, as its name suggests, is the occasional reinforcement of learning. While initial learning is slower, learning will be more autonomous and will continue even if the reinforcement stops.

Vicarious

This type of reinforcement involves learning from the experiences of others. In other words, we learn to perform those tasks we see others rewarded for. Similarly, we learn not to perform those tasks we see others punished for. A classic example is of a vending machine. If we see someone insert money and obtain two cans instead of one, we will be inclined to do the same. Conversely, if we see a person insert money but not receive anything, we will not risk the same fate ourselves.

However, with vicarious reinforcement, the learning continues until such time as a new observation is made. Returning to the vending machine, if we subsequently see another person successfully use the seemingly broken machine, we will change our knowledge and actions to incorporate this new learning.

With reinforcement, we need to adapt the type and amount of reinforcement according to the audience and medium being used. For example, vicarious reinforcement is not particularly useful for user guides and more advanced users may take exception to frequent and unnecessary reinforcement.

Interference

Frequently, existing habit families will interfere with new learning. Of course, the opposite is also true. Applying this idea to a situation in which a user guide might be used, we can use the example of a user who is proficient in using *Microsoft Word* learning to use *Corel WordPerfect*. The commands and procedures used to create and update a table of contents in *Word* may interfere with those of *WordPerfect* because the user has developed habits from using Word.

On the other hand, interference between existing habits and new learning can sometimes be positive. Returning to the idea of a *Word* user learning to use *WordPerfect*, some of the habits learned from Word will actually aid learning the new program, e.g. creating mail merges, the idea of creating templates or performing spell-checks.

3.5.3.4 Learning & Cognitive Processes

Behaviourist approaches to learning cannot account for the bulk of human learning because they do not take into account the role of cognitive processes. Cognitive approaches emphasise the cognitive processes of the individual and the social environment in which learning takes place. So rather than learning being a mechanical series of responses to stimuli, it is "mediated" by cognitive processes. With cognitive processes mediating between stimuli and the individual's behaviour, learning is said to be social (Gavin 1998:119). Furthermore, referring to a study carried out by Tolman (1948), Gavin explains that by a process known as latent learning, we unknowingly learn and assimilate information. We may never act upon this information or knowledge unless we need to or want to. This is where the expectancy of an outcome or consequences of behaviour determines whether we carry out a task or respond to learning.

According to Gavin (1998:120), this expectancy is affected by:

The Locus of Control
This locus can be internal or external. Where the locus is internal, a person believes that he or she can control his or her own destiny or fate. The locus can also be external which means that a person believes that his or her own fate is controlled by external factors. The ease and speed with which we learn depends on whether the locus of control is internal or external. We are more likely to assimilate a cause and effect if we are "responsible" for that cause.

Learned Helplessness
This is the belief that we cannot alter our fate and as a result, we give up. Gavin (*ibid.*) defines this as the *expectancy of non-escape.*

Explanatory Style
This refers to the way an individual perceives events, especially negative ones. Essentially, a pessimistic explanatory style leads to learned helplessness while an optimistic style empowers individuals and allows them to process alternative responses. We also learn by observing the actions of others. This is the same principle as vicarious reinforcement (see page 88).

3.5.3.5 *Habits and Automatic Processing*

The previous discussions of attention and models of selecting information included various models such as the early selection model, the late selection model (including the attenuator model) and capacity models. Whereas the early selection model effectively precludes the simultaneous execution of multiple tasks and the late selection model places restrictions on the type of tasks that can be performed simultaneously, the capacity model provides sufficient scope to allow for the execution of multiple, fairly high level tasks simultaneously. What is crucial here is not the nature of the task *per se*, because we can often perform two complex tasks at the same time, but rather the way in which we can perform the task. More specifically, capacity theory tells us that the performance of two simultaneous tasks depends on the amount of cognitive resources used by each task. Ultimately, this is a function of how we have learned how to do the task. Essentially, if we have learned a task "well", we need fewer cognitive processes to perform it. Over time, tasks we have learned well may become automatic, i.e. they are processed automatically.

A good way to begin understanding automatic processing and tasks is to look at habits. It would be virtually impossible for us to function without some form of habit to aid us in our day to day activities. As mentioned on page 87 "habits are learned connections between a stimulus and a response". Habits are essentially automatic tasks. They are carried out without any conscious effort on our part. Indeed, it would require significant effort for us not to develop habits and, once they have developed, to prevent ourselves from performing these tasks (Raskin 2000:20,21). Of course, tasks do not spontaneously become automatic. They require practice, rehearsal[2] and repetition. Indeed, the more a task is practised and the more familiar we become with it, the easier it becomes and the less we have to concentrate on it (Gavin 1998:33; Raskin 2000:18,19). Take for example, a student learning to play the piano. When starting to play, the student may find it difficult to play without looking at the keys. However, with practice, the student learns the position of the various keys simply by touch. Eventually, the student will be completely comfortable playing without ever looking down at the keys. Similarly, cognitive tasks can become automatic. An automatic process, therefore, occurs without any intention to do it. Such automatic tasks can even override non-automatic tasks. This can be illustrated by means of what is known as the *Stroop Effect* (Faulkner 1998:48). This effect was used to show that the act of reading is so automated, i.e. we have practised it to such an extent, that it takes virtually no conscious effort and can actually take priority over conscious tasks. The experiment involves writing the names of different colours, e.g. red, green, blue, yellow, black etc. but using different colour inks to those described by the word (see Figure 8 below).

Red

Green

Blue

Yellow

Black

Figure 8: Illustrating the Stroop Effect

[2] Rehearsal is the process of repeating information repeatedly in one's mind.

When subjects were asked to name the colour inks used to write the words, it was found that the information obtained by reading the words overruled or seriously hindered the information which came from recognising the colour of ink. This shows that for the majority of people reading has been practised to such an extent that it has become an involuntary or highly automatic task. In a sense, reading has been "over-learnt" (Gavin 1998:41) and it interferes with other tasks. In practice, however, what this means is that the process of reading requires little in the way of conscious action and few of the resources provided by the cognitive system. So what we have, therefore, is a process whereby we can process and assimilate information without expending excessive amounts of resources. This leaves the remaining resources free to be allocated to other tasks, e.g. learning and solving problems.

3.5.3.6 *Reasoning & Problem-Solving*

In the preceding discussion of automaticity and simultaneous tasks, we concentrated on the fundamental cognitive functions involved, namely attention and selection. We saw that with practice, our knowledge about how to perform tasks becomes automated and requires less cognitive effort. However, this discussion omitted a crucial fact: we do not necessarily have the skill and procedural knowledge needed to perform these tasks from the outset.

Referring back to the discussion of procedural knowledge on page 73, we know that procedural knowledge is basically knowledge about how to perform tasks. But what happens when we encounter a task for the very first time? Having never been confronted with it, we do not have a procedure for achieving our goal. This is what is termed in cognitive psychology as a problem. Coe (1996:99) defines a problem as "a goal for which we have no attainment strategy".

As Anderson (2000:240) notes, human cognition always takes place with a view to achieving goals and removing the obstacles that prevent the achievement of goals. So in a sense, virtually all human activity is either problem-solving or originated in problem-solving. Anderson (2000:241) continues to say that there is a tendency to use the term "problem" only for "original difficult episodes" but in reality, all procedural knowledge stems from the resolution of problems. The fact that some tasks become automated does not mean they are not responses to problems.

Reasoning

While problem-solving revolves around finding strategies for dealing with new tasks or experiences, it depends on our ability to reason, i.e. how we make use of existing knowledge to draw conclusions or infer something from either implicit or explicit premises. Reasoning is a cognitive process whereby we reach a conclusion and then determine whether it is valid or invalid by applying certain logical criteria (Dix 1998:38; Coe 1996:109; Ellis & Hunt 1993:290-291). There are three types of reasoning: *deductive*, *inductive* and *abductive*.

Deductive Reasoning

Deductive reasoning attempts to arrive at logical conclusions on the basis of a set of explicit premises. With this type of reasoning, we seek to derive particular conclusions from general truths which we know or believe to be true (Coe 1996:110).

As a whole, reasoning is based largely on the study of formal logic (Ellis & Hunt 1993:291) and involves reaching conclusions based on general assumptions (*ibid.*). However, there a number of points that must be remembered here. Firstly, humans do not always reason logically (Ellis & Hunt 1993:295; Hill 1995:20). Secondly, a conclusion may be correct from a purely logical point of view, but it may have no relation whatsoever to the real world or how we view it. An example of this would be the following:

If the sun is shining, it is night time.
The sun is shining.
Therefore it is night time.

Table 2: Logically Correct but Factually Incorrect Deduction

We can see from this example that a logically correct deduction is not necessarily true in terms of the real world.

Inductive Reasoning

Inductive reasoning is the process of generalising information from specific cases we have seen and then applying this general knowledge to cases we have not seen. For example, we can generalise from our past experiences that all birds have beaks because every bird we have seen had a beak. However, this method is unreliable in the sense that assumptions made using this method cannot be proved to be true – they can only be proved to be false. This is because we cannot possibly see every bird that ever lived or will live. And so there is a risk – theoretically, at least - that the next bird we see may not have a beak. However, each bird we see that *does* have a beak serves as a positive instance which reinforces our generalisation.

To put this in context, let us assume we say all cars have four wheels. Every car we have ever seen has had four wheels. Each one of the many cars we saw simply served to reinforce this generalisation until one day we see a *Robin Reliant* which has only three wheels. This challenges our belief that all cars have three wheels. However, the fact that we have seen so many examples of four-wheeled cars means that while our generalisation has been proved to be false, we are unlikely to discard it because cars almost always have four wheels. We may modify this generalisation to say that cars have three or four wheels but usually four. This illustrates the iterative nature of reasoning whereby we derive, apply and modify general truths (Coe 1996:110).

In spite of the unreliability of inductive reasoning, it serves as a useful method for maintaining information for general purposes and allowing us to make fairly stable generalisations about the world.

Abductive Reasoning

Abductive reasoning refers to the way we derive explanations from facts (Hill 1995:21; Dix 1998:40). Essentially, abduction involves us trying to find the causes of or explanations for things we see.

Let us suppose that Bob always walks to work when his car is broken. If we were to see Bob walking to work one morning, we might infer that his car is broken. As plausible and possible as this may seem, it is unreliable because Bob may simply have decided to walk to work because he wants to get more exercise.

Despite the fact that abductive reasoning is very unreliable, people frequently infer explanations using this method. Indeed, beliefs acquired using this method will persist until events occur to show that an alternative is true (compare with vicarious reinforcement on page 88).

Theoretical Approaches to Problem-Solving

There have been numerous theoretical approaches to problem-solving over the decades but the most influential approaches are, perhaps, the stimulus-response approach, Gestalt theory and the information processing approach (Ellis & Hunt 1993:287; Gavin 1998:104).

Stimulus-response Approach

This approach assumes that learners approach problems with a number of existing habits of varying strengths arranged into habit-family hierarchies (see page 87). Based on the principle of habit formation, this approach maintains that certain habits are used to tackle problems at the expense of other habits; the chosen habits are strengthened while the others are weakened. This approach, like that of the Gestaltists is internalised and does not provide enough evidence to prove reliable.

Gestalt Theory

The Gestalt theory rests on the fundamental assumption that the way people solve problems depends on how they perceive and structure their problem environment (Ellis & Hunt 1993:288). Gestaltists maintain that humans' ability to reorganise and recombine their perception of the problem allows problems to be solved. This approach identifies four stages in problem-solving (Ellis & Hunt *ibid.*; Gavin 1998:105): *preparation*, *incubation*, *illumination* and *verification*. Gestalt theorists such as Wertheimer (1959) maintain that problem-solving is both a productive and a reproductive process. Reproductive thinking draws on or reproduces previous experience while productive thinking involves restructuring, insight and the creation of new organisations or *Gestalts* (Dix 1998:44; Gavin *ibid.*).

Preparation	A person gathers information and makes initial attempts to solve the problem.
Incubation	The problem is left for a while and other tasks are carried out.
Illumination	The solution occurs to the person suddenly after incubation.
Verification	The solution is checked to see that it actually works.

Table 3: Stages in Gestalt approach to problem-solving

Gavin (*ibid.*) makes the point, however, that because Gestalt theories are based on introspection, there is insufficient proof that all of these stages occur in all cases. Consequently, such theories "lack the comprehensiveness necessary for a good theory" (Ellis & Hunt *ibid.*). While Gestalt theory ultimately proved unsatisfactory in terms of explaining problem-solving, it did prove useful because it marked a shift away from previous stimulus-response approaches (Ellis & Hunt *ibid.*) and towards the information-processing theory which is now so prevalent.

Information-Processing Approach

In contrast to both the Gestalt and stimulus-response approaches to problem-solving, the information-processing approach attempts to model problem-solving from a computer perspective. The aim is to formulate steps and rules which are involved in solving problems in order to produce an abstract model of the process (Ellis and Hunt 1993:289). Developed by Newell & Simon (1972), the information processing approach places problems in what is known as a *problem space*. In his book written with Card and Moran, Newell defines a problem space as

> ...a set of states of knowledge, operators for changing one state into another, constraints on applying operators and control knowledge for deciding which operator to apply next. (Card *et al.* 1983:361)

We can further explain the problem space concept by saying that it consists of various states of a problem (Anderson 2000:242; Dix 1998:41). A state in this regard is a particular representation of the problem. In solving problems, a person starts out from what is called the *initial state* where the problem is unsolved and navigates through the problem space until the *goal state*, where the problem is solved, is reached (Anderson *ibid.*; Gavin 1998:106; Dix 1998:41-42).

In moving from the initial state, the person changes one state into another using problem-solving operators. Operators are possible moves which can be made in order to change one state into another or to divide goals into sub-goals. Basically, problem-solving involves finding a series of operators which lead from the initial state to the goal state. One crucial feature of the problem space model is that it takes place within the cognitive processing system and as such, is limited by the capacity of STM and the speed with which information can be retrieved from LTM. It is also important to note that there are different problem spaces for different tasks and that problem spaces may change over time as a person becomes more familiar with the task (Card *et al.* 1983:87). As states are converted into other states, it may be possible to use any one of a number of possible operators. The challenge here is to select the appropriate one to form one of a series of stages which make up problem-solving.

Stages in Problem-Solving

In the information-processing approach, a problem is placed in a problem space. We know that the problem space consists of the various states of the problem, namely the initial and goal states as well as the states in between. But how is it that we arrive at the goal state? To understand this, we need to examine the various stages in problem-solving.

1. Identifying and Understanding the Problem
2. Devising and Selecting a Strategy
3. Carrying out the strategy
4. Checking whether the strategy actually worked.

Much of the difficulty encountered in problem-solving stems from a failure to fully comprehend the problem and to recognise its features. If we do not fully understand a problem, we cannot hope to solve it (Ellis & Hunt 1993:266). Consequently, the way we perceive or "frame" the problem ultimately plays a decisive role in our success (Coe 1996:48). There are four principal types of knowledge which affect the way we interpret a problem (see also Section 3.4.3.1 on page 71).

Factual knowledge	Consists of rules, categories and representations of the problem
Semantic knowledge	Conceptual understanding of the problem
Schematic knowledge	Provides an infrastructural understanding of the problem; the various related issues and factors and how they relate to each other
Strategic knowledge	An understanding of how to build strategies for solving problems within the overall problem area

Table 4: Types of knowledge affecting the understanding of problems

Once we have identified and understood the problem, the next stage is to formulate and select a strategy for solving the problem. In this stage, we attempt to find a possible solution to a problem. Indeed, we may even formulate several possible solutions to a problem (Ellis & Hunt 1993:267). Again, we have several strategies for devising solutions.

RANDOM TRIAL-AND-ERROR

This process involves the random selection and implementation of operators until the goal state is achieved. Naturally, such an approach can be extremely inefficient and time-consuming

HILL CLIMBING

This involves gradual movements away from the initial state and towards the goal state. This approach can result in misleading choices and increase the time needed to reach the goal state.

HYPOTHESIS TESTING

Hypothesis testing is similar to trial-and-error but instead it is a purely cognitive activity. Rather than testing a large number of random possible solutions, a smaller, more considered range of solutions is tested. Knowledge of the problem area is used to restrict the selection of possible solutions to ones which have a realistic chance of success.

ALGORITHMS

These are sets of precise rules or procedures which guarantee the successful solution of the problem. Algorithms take a set of rules and processes that have worked in the past and apply them to the current problem.

HEURISTICS

Heuristics are similar to algorithms but differ in the fact that they do not guarantee success. Rather, they are general guidelines or "rules of thumb" (Ellis & Hunt 1993:267; Coe 1996:123). They are loose sets of procedures which can assist in solving a problem. One useful heuristic is the means-end analysis. This is a complex process whereby a person works on a single goal at a time. If it is not possible to reach the goal state at present, the goal is divided into sub-goals with the aim of removing the obstacles to the achievement of the primary goal (Anderson 2000:253; Gavin 1998:103).

Once we have selected and implemented a solution, the next stage is to see whether it worked. Here too, a clear understanding of the problem and also the solution is needed in order to determine the success of the solution. This essentially involves some form of judgement as to the effectiveness of the solution.

Difficulties Involved in Problem-Solving

Despite the wide range of strategies and processes involved in problem-solving, the seemingly straight-forward task of solving a problem is not without its difficulties. One of these difficulties is known as a *problem solving set*. This refers to the fact that we have a tendency to view problems through our own experience which prevents us from looking at problems in novel or inventive ways. *Confirmation bias* refers to our tendency to search for solutions that confirm our view of existing ideas. We are reluctant to accept ideas which are different to or which challenge our beliefs. *Functional fixedness* refers to our inability to see the flexibility of objects' functions. Finally, if we think back to our discussion on habits (see page 87) we can see that our past experiences can play a part in how we solve problems. *Negative transfer* is the phenomenon whereby existing habits and experiences inhibit or prevent us from solving problems.

3.5.3.7 The Transition from Problem-Solving to Skill

While we are learning, we are constantly solving problems. When reading a user guide, for example, we perceive new information and we process it in order to understand not only the instructions in the text but how they relate to the task at hand. Indeed, all new tasks start off as problems which we need to solve (Anderson 2000:240-1). As we become more proficient and knowledgeable, the extent of problem-solving diminishes as we learn how to perform tasks and these tasks become automatic. As tasks become increasingly automated, they require fewer cognitive resources and can be performed more quickly (Raskin 2000:20-21).

Preece (1994:164-5) maintains that the transition from problem-solving to skill involves converting declarative knowledge into procedural knowledge or skills. However, in order to perform this conversion, users need sufficient declarative knowledge before the associations can be made. Only when this repository of declarative information is present and the associations established between items of information, can a task become automated or procedural. The course of this transition from problem-solving to skill can be mapped by means of user curves.

User Curves

User curves are a way of visualising the progress of learners and helping to determine their needs and achievements. Coe (1996:44ff) compares a user's approach to learning how to use new technology with Erikson's stages of development (Erikson 1965: 239-266). The *Technical Information User Curve* is introduced to help plot the stages in learning technical information from entry level users to power level users.

Entry-level users perform only what they are instructed to and achieve the direct results of those instructions. At this early stage, users can only comply with direct instructions and they are only concerned with their immediate circumstances.

Beginners focus on the basic aspects of the task at hand and begin to build simple symbolic models. They cannot yet apply these models elsewhere.

Intermediate users learn more about the basics and develop an understanding of the logic and principles involved.

Power users have a deep understanding of the product and can apply this knowledge to new tasks.

Another unique type of user is the *outsider* who does not participate in the learning process. Outsiders could be occasional users who do not want or need to learn how to use the product, preferring instead, on the rare occasions that they need to use the product, to learn a single task well enough to perform the immediate task. This *ad hoc* use and learning cannot be charted on the curve.

Users can approach this learning curve at any stage and, with practice, they can move up through the various stages. They also can choose not to participate in the curve and remain at the outsider stage.

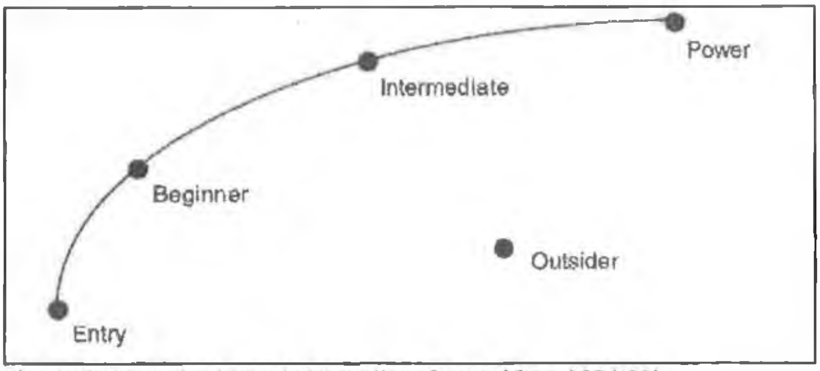


Figure 9: Technical User Information Curve (Coe 1996:51)

Although Erikson's Psychosocial Curve aids visualising a person's social development, it can also be applied to user learning in conjunction with the *Technical Information User Curve*. The stages on this curve are:

Trust

Users need to trust in the information and be confident that it will work. Otherwise users are likely to stop there and then (cf. Schneiderman 1998:12-13). At this stage, users follow instructions to the letter. For users to proceed beyond this stage, they need to have faith in the instructions and be confident that they will have the anticipated outcome. Users need constant reassurance and need to feel that their environment is predictable and reliable.

Self-confidence

Users begin to feel less exposed and more confident as regards their abilities. They come to have more faith in the information contained in a user guide. This is a critical stage in the development process because if at any stage the trust built up is broken, i.e. users are given incorrect or faulty information, the user will lose confidence in the instructions and the product and may not use either again.

Initiative

At this stage users are confident and knowledgeable and can comfortably use the product without assistance, prompting or recourse to the user guide. The user guide becomes a useful but non-essential source of information which users can refer to if necessary.

Competence

With more experience, users become more confident and begin using knowledge for different tasks and applying skills in different contexts.

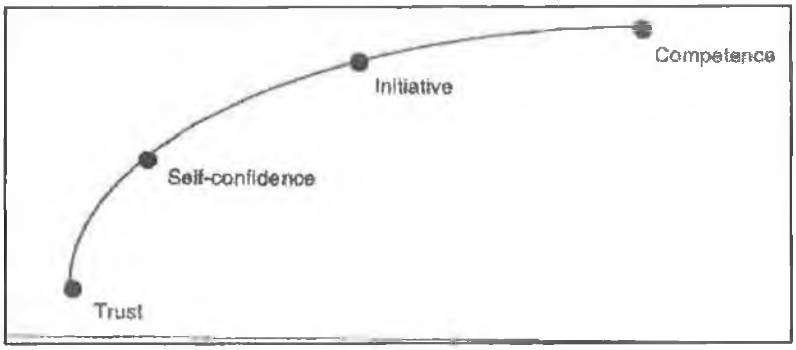


Figure 10: Erikson's Psychosocial Curve (Coe 1996:53)

If we compare the two curves we can see the psychological states and information requirements of users as they progress from novice to expert. The aim of a user guide is, therefore, to guide users along these curves through the various stages of learning. In order to do this, the user guide must take into account the various cognitive factors which affect how users process and store knowledge.

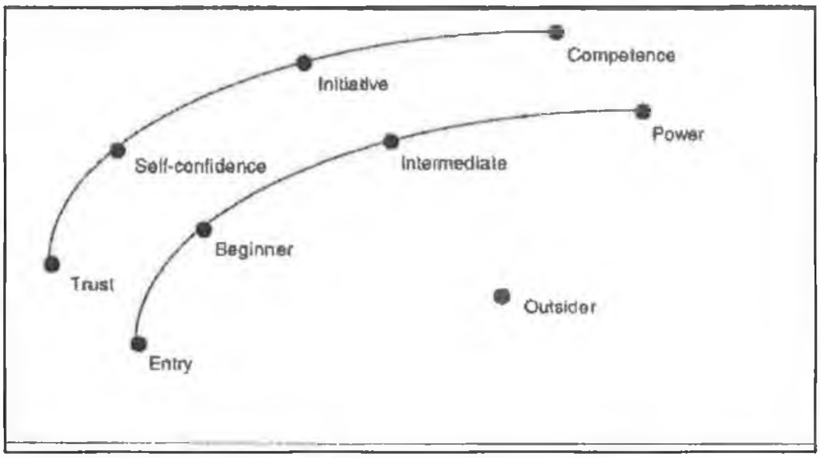


Figure 11: Comparison of Erikson's Curve and the Technical Information Curve (Coe 1996:53)

3.6 CONCLUSIONS

This chapter began by defining usability as the *ease* with which users can perform tasks *effectively* and *efficiently* and their level of *satisfaction* with their work. Having discussed the various factors which contribute to usability, we set about understanding the nature of usability. Usability means that products must take users' mental abilities and limitations into account. To understand usability, it was necessary to explore the various systems and processes that make up human cognition. Human cognition is likened to an information processing system in that humans receive data or information from their surroundings which they then process. These processes enable humans to make sense of the information they require and to decide what, if anything, can be done with it. If a response is necessary, our cognitive processes help us select the appropriate course of action. Sight, the human sense which provides the most important sensory input in relation to user guides, was discussed and its capabilities and limitations were outlined.

We have seen that sensory input is subjected to several, often complex processes in short-term memory (STM). It is here that we decide on courses of action and from here that information passes on to long-term memory (LTM). Armed with this knowledge, we can see how it is that printed words are converted into information that we can use and how this information needs to be processed in order for us to remember or learn it. Of course, the flow of information from STM to LTM is not automatic, guaranteed or even efficient. The chapter outlined the broad phases involved in turning ephemeral information stored in STM into lasting knowledge stored in LTM through learning. Some of the various obstacles to this process were also presented.

This chapter provided us with the foundations upon which we can present a more detailed discussion of usability and how we can take human cognitive abilities and limitations into account to ensure user guides are as usable as possible. The task facing us now is to take this knowledge and put it to good use in improving the usability of user guides.

Chapter 4
USABILITY & COGNETICS

4.1 INTRODUCTION

> Accommodating the diverse human perceptual, cognitive, and motor abilities is a challenge to every designer...the presence of a computer is only incidental to the design; human needs and abilities are the guiding forces. (Schneiderman 1998: 18)

This chapter examines the field of cognitive engineering or cognetics. This is the study of adapting "products" or interfaces to suit humans while making their work as simple and untaxing as possible. In essence, cognitive engineering allows us to put our understanding of human cognition into practice. The chapter will discuss how an understanding of cognition can be introduced into design processes to make interfaces more usable and look at ways of ensuring that users can work with a product effectively and efficiently. It will first be necessary to define what we mean by interface and then apply this definition to texts. After examining some principles of interface design and how to implement them, we will then explore ways of examining user guides from an interaction point of view to identify areas for improvement.

This chapter will then introduce Iconic Linkage (IL) as one method for improving the usability of user guides. The chapter will present the origins and nature of IL and proceed to discuss the potential benefits of IL and how it can improve usability.

4.2 INTERFACES

When we speak about *user interfaces* many people assume we are referring specifically to the graphical user interfaces (GUI) of modern computers. While GUIs are perhaps one of the most common and recognisable types of interface, they are precisely that – types of interface. The reality is that not all interfaces have windows, icons and menus: interfaces can be found on VCRs, mobile phones, digital watches, ATM machines and even microwave ovens. It is very easy to give examples of interfaces but actually defining interfaces is another matter. Card *et al.* (1983:4) state that it is easy to locate the interface between computer and human simply by starting at the CPU and tracing "a data path outward... until you stumble across a human being". This, however, by the authors' own admission is less than clear and we are left with no real clue as to the nature or boundaries of the interface.

Faulkner (1998:54) maintains that the human-computer interface mediates between the user and the computer system. Again, this is somewhat vague. Perhaps we should look to the function of the interface in order to understand what an interface is. Bødker (1991:77) proposes that "the basic role of the user interface is to support the user in acting on an object or with a subject". She continues by saying that a good user interface allows users to focus on the task at hand rather than on other objects or subjects. So, like the software system itself, the purpose of interfaces is to allow us to do something – in this case, to use the system. In other words, an interface is a tool or a means to an end. Such a view is echoed by Raskin (2000:2) who defines interfaces as "the way that you accomplish tasks with a product".

Perhaps one of the clearest and most useful definitions of an interface is that provided by Preece (1994:13):

> "The user interface of a computer system is the medium through which a user communicates with the computer. [...] the user interface can be thought of as those aspects of the system with which the user comes into contact both physically and cognitively" (Preece 1994:13).

Admittedly, the above definition of interfaces is rather vague in terms of concrete physical details but it is sufficiently detailed in terms of function to allow for variations in the physical characteristics of interfaces and their areas of use (as mentioned above). This flexibility is essential when we consider the case of software user guides. Ostensibly, the purpose of such guides is to teach users how to use a software product. Without such training, many users would not be able to use the software; although a few may try to use it by a process of trial and error although they are likely to have to be less efficient in their learning than those who use the user guide. In other words, user guides facilitate the use of software products and in a sense become part of the human-computer interface. If we were to be very specific, the user guide would be an interface between the user and the software's interface but it is more convenient to simply regard it as an extension of the overall human-computer interface.

4.3 Cognetics

Ergonomics is the design of machines to take into account the physical variability of humans. For example, we know that a human cannot possibly be expected to press two buttons positioned three metres apart (Raskin 2000:10). With our knowledge of the human body and the standard level of variation among different humans, we engineer our physical world to suit our needs and capabilities. Similarly, we need to engineer our world to conform to our mental capabilities and limitations. Essentially, what we are talking about is an ergonomics of the mind. This is known as *cognitive engineering* or *cognetics.* In reality, cognetics is a branch of ergonomics but the term ergonomics is used primarily to refer to the physical aspects of human-orientated design.

A key factor which is frequently overlooked by software designers, engineers and even users is that computers are supposed to be tools which assist humans in doing something else. Computers should, therefore, reflect the needs, capabilities and limitations of the humans who use them. As Faulkner (1998:2) says, a computer "has to play up to [users'] strengths and compensate for their weaknesses". Raskin (2000:10) maintains "you do not expect a typical user to be able to multiply two 30-digit numbers in five seconds and you would not design an interface that requires such an ability". But this is an obvious example. Other factors are more subtle and relate to the way we perceive and process information, solve problems, learn and access knowledge – even how we read.

The main challenge facing software manufacturers is to produce systems which people really want, need and can use despite the complexity of the task being performed (Faulkner 1998:129). While decisions as regards what people want and need from a product are usually based on economic factors and made by people other than the actual system designers, ensuring the usability of systems remains the primary focus of cognetics.

Referring back to Section 2.3.1 we have seen that efforts to ensure usable and successful documentation need to be made from the very start. It would be extremely helpful if the obstacles to usable documentation are overcome at the beginning of the development process. With this in mind, it would be useful to examine the way in which interfaces are designed and developed. We will first look at the goals which must be achieved in order for an interface to be regarded as usable. We will then examine the development processes which form the environment or context for the various design strategies used in cognitive engineering.

4.3.1 Usability Objectives

Faulkner (1998:130-131) maintains that to ensure that a system is as usable as possible, there are three fundamental goals which need to be achieved: *learnability*, *throughput* and *user satisfaction*.

Learnability refers to the time required to learn the system or to reach a specific skill or performance level. This objective can be quantified by examining the frequency of errors, the type of errors made etc. Dix (1998:162) expands this category by including the sub-headings of predictability, familiarity, generalisability and consistency. Familiarity refers to the way information presented relates to users' existing knowledge which they bring with them to the learning process. Generalisability relates to the ability of users to use the information learned in other situations.

Throughput refers to the ease and efficiency of use after an initial learning period. This is quantified by examining the time needed by users to perform tasks, their success rates when performing tasks, the time spent looking for help etc.

User satisfaction is a subjective goal but it can give an overall picture of how well the system performs in the eyes of users. This can be quantified by asking users to fill out a questionnaire rating aspects of the systems performance etc., for example on a scale from (1) very bad to (5) very good.

Schneiderman (1998:15) adds an additional goal which he terms *retention over time*. This is particularly relevant to user guides in that their purpose is to teach users and facilitate their use of the system. Retention relates to how well users maintain their knowledge over time as well as the time needed for learning and frequency of use.

Quesenbery (2001) provides a similar taxonomy of usability characteristics which she terms the "*5 Es*". For a system to be regarded as usable it must be:

- effective
- efficient
- engaging
- error tolerant
- easy to learn

Effective

Effectiveness refers to the ability of users to perform tasks completely and accurately. If a user's goals are met successfully and if the user's work is correct, the system is regarded as usable. Quesenbery (*ibid.*) states that effectiveness can sometimes be confused with efficiency. She clarifies this by saying that where *effectiveness* relates to *how well* a task is carried out, *efficiency* deals with *how quickly* a task is carried out. An effective user guide should, therefore, provide correct information which will allow the user to complete tasks successfully.

Efficient

Efficiency is the speed at which users can perform their tasks accurately. In the ISO 9241 standard, efficiency is defined as the total amount of resources expended in performing a task. Quesenbery makes the point that in order to fully appreciate efficiency, it is essential that the notion of a task be approached from the point of view of a user. Whereas a system designer might treat an interaction between a user and a system as a series of small tasks, these tasks are grouped together by users to form one large task. Thus, the procedure for connecting a new printer to a computer might consist of small, efficient tasks such as connecting the cable, installing the drivers, calibrating the printer etc., but if the total time needed to complete all these tasks is greater than that the amount of time a user is prepared to spend, the overall performance of the system is inefficient. In the case of user guides, the information should be clear, digestible and concise. If users have to grapple with unwieldy constructions or ambiguous instructions, the efficiency of the system will undoubtedly suffer.

Engaging

An interface or system is engaging if it is pleasant and satisfying to use. An engaging system will hold a user's attention and make using it a rewarding experience. This characteristic can be affected by such things as the visual design of the interface where the readability of the text as well as the type of interaction can change a user's relationship with the interface and system. The way in which information is chunked also plays a role in how engaging an interface is – as mentioned in Chapter 3, the way information is chunked helps maximise the resources of a user's short-term memory.

Error Tolerant

In an ideal world, every system and interface would be free from errors and the people who use them would not make any errors. However, it would be naïve to expect users not make at least *some* mistakes. Consequently, a usable system should pre-empt the types of errors a user is likely to make and either make it very difficult to make these errors or at least provide ways of identifying and rectifying errors when they occur. In the case of user guides, clear and precise information and instructions are essential. Similarly, warnings and precautions should be given in good time to prevent users "jumping the gun" or performing potentially invalid actions.

Easy to Learn

Like the idea of learnability described on page 110, ensuring that a system is easy to learn should allow users to quickly become familiar and confident with the fundamental workings of the system and provide them with a basis for continued learning. A common barrier for users is the typically steep learning curve required for new users. Making systems easy to learn, Quesenbery (*ibid.*) maintains, involves more than just presenting information in a way that is easy to understand. Ideally, users should be allowed to build on their prior knowledge which they bring with them to the interaction (see *Constructivist Approaches* on page 64).

Similarly, if we make the interface as consistent as possible, it should be possible for users to re-use interaction patterns they acquire. This serves to reinforce the information they have already learned as well as that which they will learn. Consistency and predictability are also key factors in Quesenbery's description of the ease with which users learn to use systems. If we use familiar terminology and functionality, users develop expectations as to how the system will behave and this in turn inspires confidence in the user.

We can see that usability is much more than just making sure that everything works with as few problems as possible. It requires a deep understanding of users, what they want to do and what they are capable of doing. Similarly, making sure a product is usable is much more than "tightening bolts, smoothing off rough edges and applying a coat of paint". Rather, it is much more involved and is a complex process. According to Dumas & Redish (1993:5) "usability means focusing on users" and in order to develop a product that is usable, it is essential to know and understand people who represent real users.

So how do we set about making systems usable? We need some way of engineering them to make sure that they take into account the problems faced (and posed) by users, particularly their cognitive capabilities. The following sections will discuss a number of interface development models and explore ways in which usability can be engineered into interfaces.

4.3.2 Interface Development Models

Computer systems are frequently very complex and, as a result, the process of developing them is often equally as complex and can require vast amounts of information from numerous different sources. The following paragraphs briefly describe some of the more common software development models. In describing these models, it should be remembered that they can apply to interfaces in general and to user guides in particular.

4.3.2.1 Waterfall Model

According to Landauer (1995:172), "most software development follows a sequence of activities called the 'waterfall model'". This model is a linear, hierarchical model which begins with the definition of the software system and of the software requirements. With this information in hand, a requirements specification is produced and this forms the basis of all work in the development project. The product is designed on the basis of the requirements document and this describes in detail all of the aspects of the system to be produced. Once the design has been established, the implementation phase sees the design put into practice. This stage involves programming and testing the various components of the system, the production of user documentation, online help and so on.

When all of the components have been completed, they are pieced together to form the finished product ready for verification and acceptance by the client. Maintenance will then be carried out on the deployed product and this may result in changes being made to the product. (Preece 1996:356).

At each stage of development, work completed is subjected to what is known as *Validation, Verification and Testing* (VV&T). This ensures that the product conforms to the product specification, that it is consistent with the previous design concept and that it actually works. The product must successfully pass VV&T before it can proceed to the next stage of development. The benefit of this model is that it allows for efficient and close supervision of progress by development managers. However, this is where the benefits end. Landauer describes the waterfall model as follows:

> It starts upstream with requirements that feed a tumbling torrent of planning and programming that – usually – plunges into chaos just before it comes to a rest (Landauer 1995:173)

More specifically, Preece (1994:357) highlights the following problems with this model:

- The product specification is often written in vague, generic and ambiguous language and can make demands formulated by managers which are not always technically feasible.
- Many projects are initiated at board level and are then imposed on user departments regardless of their capabilities or needs.
- Maintenance is one of the most important stages in development and also one of the longest. This becomes a problem if the development team is disbanded after the product is released. This is quite common according to Preece.
- The model does not recognise that organisational changes may occur as a result of the introduction of new software.

Of all these problems, Preece maintains that the first one mentioned is the most serious because of the potential ambiguity of the product specification. This can be as a result of either imprecise language or because of a lack of in-depth understanding of what is technically realistic or feasible. She continues to say that "it is really impossible to completely understand and express user requirements until a fair amount of design has been undertaken" (*ibid.*).

4.3.2.2 Spiral Model

The spiral model (Boehm 1996) of software development is specifically aims to identify the most significant risks to the success of the of the design at any given stage in the development process. It is based on the concept of prototyping whereby parts of the software system are produced in working model or prototype form. These prototypes mean that the product design can be tested at each step of the way to ensure that they comply with the user requirements.

This model, while better in terms of the ability to test the product at various stages of development, still utilises key elements of the traditional development process outlined above, namely requirements gathering, design and implementation. What makes it different is the numerous iterations (versions or prototypes) of the product throughout the development process. As each iteration is produced, it is tested and validated to see what changes and improvements are needed. These changes are implemented in a new prototype which in turn is tested and modified. However, the more iterations are required, the greater the costs and time required to finish development.

In an effort to curb the apparent problem of excessive costs and delays, the 'W' model was developed. This model involves the creation of a single design implemented on a small scale. This is then tested and the changes implemented in the overall design. Development then proceeds using the traditional development approach.

4.3.2.3 Logical User-Centred Interactive Design (LUCID)

Another method which incorporates the iterative approach to development and design is the LUCID model developed by Charles Kreitzberg (Kreitzberg 1998). This method involves six stages of software development:

Stage 1	Envision (Develop a product concept)
Stage 2	Discovery (Perform research and needs analysis)
Stage 3	Design Foundation (Design concepts and key-screen prototype)
Stage 4	Design Details (Do iterative design and refinement)
Stage 5	Build (Implement software)
Stage 6	Release (Provide rollout support)

Stage 1
The purpose of this product concept development stage is to develop a clear, unambiguous and shared vision of the product. During this stage a "high concept" (Schneiderman 1998:106) is created for the product. This is a brief statement which defines the goals, functionality and benefits of the product. Once the product concept has been defined, any environmental, technical or legal constraints which affect the product are identified. Next the user population is specified and usability goals are established. A project plan and budget are also prepared. The prototype for this stage consists of simple paper-based sketches of what the product's screens will look like.

Stage 2
The *Discovery* stage aims to investigate and understand how users interact with the tasks and/or information. With the concept team and project team in place, a detailed analysis of the intended users is carried out to determine their usability needs as well as the tasks they will perform. The business processes the product is intended to support are also examined. Working in conjunction with representative users, the project team designs workflow scenarios and defines objects which are central to the product's design. Checks are also carried out to ensure that the design architecture is supported by the system architecture.

Stage 3
The *Design Foundation* stage involves designing the key elements of the interface design. During this stage a "key-screen prototype" (Schneiderman *ibid.*) is produced using prototyping tools to show users the design of the proposed system. The aim here is to introduce the system to the users and obtain feedback from usability evaluations to determine whether changes need to be made or whether new elements need to be added to the design. This stage involves creating usability objectives based on stated user needs and creating guidelines and style guides for implementing the

design as well as a navigational model to determine how users move through the system.

Stage 4

The *Design Details* stage involves refining the interface design. During this stage a full, working prototype is created from the key-screen prototype. The working prototype is then subjected to a series of heuristic and expert reviews to ensure that it functions correctly and to ensure that it meets the product requirements. Next a series of full-scale usability tests are carried out. Feedback from these reviews and tests is incorporated into the prototype.

Stage 5

The *Build* stage involves taking the full, working prototype developed in the previous stage and using it as the basis for the programming specification. This is where the design and prototypes are put into practice. At this point, standard implementation practices are formulated and any late changes are incorporated if necessary. Usability tests that require working program code are carried out during this stage. In addition, all user assistance materials such as online help, documentation etc. are produced during this stage.

Stage 6

The *Release* stage involves the phased rollout of the finished product. In addition to ensuring the usability of the final product, pre-installation support is provided to ensure the smooth adoption of the product. Training and assistance is provided and any problems which occur are logged. The product is evaluated to prepare for the next version of the product. Maintenance and consultation are undertaken on a continuing basis.

After each stage in the development process, the progress of the project is evaluated with regard to 12 distinct areas, each of which is linked to specific deliverables.

1. Product Definition	High concept for management
2. Business Care	Pricing, expected revenue, return on investment, competition
3. Resources	Duration, effort levels, team actions, backup plan
4. Physical Environment	Ergonomics, physical installation
5. Technical Environment	Hardware and software for development and integration
6. Users	Multiple communities for interviews, user testing, marketing
7. Functionality	Services provided to users
8. Prototype	Paper, key-screens, running prototypes
9. Usability	Set measurable goals, conduct tests, refine interface and goals
10. Design Guidelines	Modification of guidelines, implementation of review procedure
11. Content Materials	Identification and acquisition of copyrighted text, audio and video
12. Documentation, Training & Help	Specification, development and testing of paper, video and online versions

Table 5: Evaluation Criteria in LUCID Framework

The advantage of this method is that it explicitly incorporates usability and indeed attaches major importance to it in the development process. This framework specifically encourages iterative development and according to Kreitzberg, many of the tasks in each stage are iterative in so far as they are "repeated in a rapid cycle with review tasks until the result is a satisfactory conclusion" (Kreitzberg *ibid.*).

While the above paragraphs are by no means exhaustive in their description of development processes, it is clear that successful design processes involve some form of iterative development. The design of the product is firmly embedded in the process and as a result, the design cannot be perfected in one single attempt (Landauer 1995:173). With such obvious importance being attached to the design of the product, the following sections will examine the processes and methods used in the design of software products. Again, because development and design are so closely linked, it is difficult to separate the two areas and avoid overlap or even repetition of tasks and responsibilities. However, for the sake of clarity and simplicity, I will attempt to make a broad division between the two.

4.3.3 Design

Before we can build an interface, we must first design it. While this may seem axiomatic, it is experience tells us that not enough time and effort is spent on designing interfaces before development begins (Landauer 1995:217-8). Preece (1994:352) maintains that design refers to both

> the process of developing a product, artefact or system and to the various representations (simulations or models) of the product that are produced during the design process.

We can see from this that design is a process which is constantly undergoing some form of change (Schneiderman 1998:99). Interface design involves the development of several solutions – which may only be temporary solutions or even partial solutions. Design is, therefore, a creative and unpredictable activity which involves a wide range of factors.

If the needs of users are to be taken into account in the design of interfaces, it is vital that users are involved to some extent throughout the design process. Preece (1994:371) comments that "users are not there simply to comment on a designer's ideas. They should be intimately involved in all aspects".

Of course this ideal is open to different interpretations. Does it mean users should be part of the project team from the outset? Or perhaps they should be encouraged to put forward their own ideas on how the product should be designed? Should users simply provide the basic idea or background for the design process or should they be involved in examining and refining the system? This subject could be argued and discussed at great length but such a course of action would be of little or no use here. What is of genuine importance is what exactly are the users' requirements.

4.4 Gathering Design Requirements

A fundamental factor of user-centred design is the need to find out what it is that users need from a system. This does not mean what functionality users would like to see implemented. As we have already established (see page 53), functionality or usefulness is not the same as usability. Rather, we are interested in the factors which will improve the interaction between the users and the system. We need to establish what will make the system easier to use and learn. In the following sections we will examine several methods for gathering user requirements for the purposes of designing a usable system and interface.

In establishing the requirements of users, there are two broad approaches: *user requirements modelling* and *task analysis.*

4.4.1 User Requirements Modelling

It is clear from the preceding sections that gathering user requirements for a system is a critical component for all design processes (Dix 1998:223). User requirements modelling (URM) is used to ensure factors such as usability and acceptability are included in the design of a system. There are several methods for conducting URM but they can be broadly categorised into *socio-technical models, soft systems methodology* and *participatory design.*

4.4.1.1 Socio-technical Models

Socio-technical models examine the broader technical, social, organisational and human aspects of design. Acknowledging the fact that a system is not just an isolated artefact, this type of URM examines the effects and ramifications of a system by examining the social and technical issues together (Dix 1998:234). Some of the more common socio-technical models include: *User Skills & Task Match (USTM)*, *Open System Task Analysis (OSTA)* and *Effective Technical & Human Implementation of Computer Systems (ETHICS)*.

User Skills & Task Match (USTM)

USTM uses the idea of stakeholders to examine the effects and requirements of a system.

- Primary stakeholders are the people who use the system
- Secondary stakeholders do not directly use the system but either input information to it or receive information from it
- Tertiary stakeholders are neither primary nor secondary but nonetheless are affected by the success or failure of the system
- Facilitating stakeholders are involved in the design, development and maintenance of the system

With this model, the design is evaluated and adapted on the basis of the requirements of each type of stakeholder.

Open System Task Analysis (OSTA)

The OSTA model seeks to describe the effects of introducing a system into an organisational work environment. This model examines the social aspects of the system (e.g. usability) together with the technical aspects of the system (e.g. functionality). There are eight stages in this model:

1. Development of primary tasks to be supported by the system in terms of user goals
2. Identification of task inputs to the system
3. Description of external environment (physical, economic and political)
4. Description of transformation processes in systems in terms of actions performed on or with the objects
5. Analysis of social system with regard to existing workgroups and relationships inside and outside the organisation
6. Description of technical system
7. Establishment of performance satisfaction criteria (including social and technical requirements of the system)
8. Specification of new technical system

Effective Technical & Human Implementation of Computer Systems (ETHICS)
ETHICS also examines the social and technical requirements for a system but uses two different design teams to deal with each group of requirements. The two teams work independently of each other and then attempt to merge their findings.

Each of these models seeks to combine the human factors of a system with the technical factors of the system. However, their treatment of human or social factors is superficial to say the least. There is no mechanism for precisely pinpointing what needs to be in place in order to meet a particular requirement. Nor for that matter is there a mechanism for identifying factors not related to the organisational nature of the tasks being dealt with. Furthermore, none of the systems specifically target the interface as being central to a user's experience with the system.

4.4.1.2 Soft Systems Methodology

In contrast to the socio-technical models, the Soft Systems Methodology (SSM) views technology and people as components of the same system. SSM is more concerned with understanding the situation in which a system is used rather than with finding design solutions (Dix 1998:227). SSM comprises 7 stages:

1. Recognition of the problem and the start of analysis.
2. Detailed description of the problem to develop a rich picture. This includes the stakeholders, their tasks and groups as well as the organisational structures and processes.
3. Generation of root definitions of the system. These include:
    - Clients who receive output or benefit from the system
    - Actors who perform activities in the system
    - Transformations, which are the changes affected by the system
    - Weltanschauung (or world view), which is how the system is perceived in a particular root definition
    - Owner of the system
    - Environment, or the world in which the system operates and which affects the system
4. Devise conceptual model. This defines what the system has to do in order to fulfil a root definition.
5. Compare the actual system with the conceptual model
6. Determine which changes are needed.
7. Decide upon required actions

The problem with SSM is that it is complex and requires much practice in order to be used effectively. It cannot provide definitive answers; there is no "right" or "wrong" answer.

4.4.1.3 Participatory Design

Also known as the *Scandinavian approach* because of its origins in Scandinavia, participatory design involves users throughout the design process, not just in the evaluation stage (Schneiderman 1998:109; Dix 1998:229; Preece 1994:375). In this approach, users are full members of the design team and as such, they play a central role in actually developing the interface design.

The reasoning behind this is that users are regarded as experts in the tasks and work to be supported by the system and as such they are ideally placed to advise designers. Indeed, the proposed system will also have an effect on the way users work so involving them in the design process is a good way of highlighting potential problems and also of ensuring acceptance of the final system. The three primary features of participatory design are that it aims to improve the work environment, it involves user collaboration at every stage of the design process and by using iteration, it allows the design to be evaluated and revised at each stage.

Participatory design uses a wide range of methods aimed at bridging the gap between user and designer. These include *brainstorming* where all members of the design team "pool" ideas in an informative yet structured way. The information and ideas gathered as a result of such a session are recorded to form a stock of design ideas.

- *Storyboarding* is a way of describing and documenting a user's day-to-day activities and for illustrating possible designs and the effects these designs will have.
- *Workshops* provide a forum for designers and users to meet and discuss issues related to the design, tasks and work environment.
- *Pencil and paper exercises* provide a fast and simple way of trying out design ideas using paper mock-ups of the interface to simulate how users would use the system.

Interestingly, this approach has not been widely implemented to any significant extent even though some of its methods and strategies have been adopted by other approaches. This may be because of the perceived expense and delays in development which may result from the involvement of users. Other criticisms levelled at the approach include possible implications for the organisational structures and work practices which may arise as well as personal problems associated with the rejection of some users' ideas. In addition, there is also the possibility that designers may have to compromise their designs in order to please users (Schneiderman 1998:110; Preece 1996:378).

The models described above are more systems-based in that they examine more than just the tangible design of the system/interface. Also, they require considerable amounts of work, planning and resources not to mention direct access to users. In practice, however, it may not always be possible to use these methods for the production or translation of user guides. With this in mind it would be useful to look at what it is users do as well as focussing on wider socio-technical or subjective issues.

4.5 Task Analysis

Task analysis is "the study of what a user is required to do in terms of actions and / or cognitive processes to achieve a task" (Nectar 1999). In contrast to the requirements gathering models outlined above, task analysis is concerned with what people do when they perform tasks. Task analysis features heavily both in the design and the evaluation phases of usability engineering. When used in the design phase, it is used to predict where potential problems and difficulties may occur.

The term *Task Analysis* itself is problematic in that it refers to a "bewildering range of techniques" (Preece 1994:410). Some techniques are designed to find out precisely what people do in terms of work and tasks, represent those tasks and predict where users will encounter difficulties and evaluate the design against usability requirements while others are intended to predict performance, learnability or complexity. A central feature common to most, if not all task analysis methods is the idea of breaking goals and tasks down into sub-goals and sub-tasks. The extent to which goals and tasks are broken down or decomposed is referred to as the granularity and this varies from method to method.

According to Preece (1994:411), part of the difficulty encountered in any examination of task analysis is the lack of homogeneity in the terminology used by different authors. Indeed, in previous chapters the term task has been used to refer to high-level "jobs" that users carry out. This is in contrast with the definition of tasks in the context of task analysis. So in order to begin discussing task analysis, we must first acquaint ourselves with the three core concepts in task analysis: *goals*, *tasks* and *actions* (*ibid.*).

- *Goals* are sometimes referred to as external tasks and are defined as "a state of a system that the human wishes to achieve" (*ibid.*). We can look at goals as being the wider objective of the human, i.e. what the user ultimately wants to achieve.
- *Tasks* or *internal tasks* are the activities which are required or believed to be necessary by the user in order to achieve a goal. Tasks are structured sets of activities which are broken down into smaller components of tasks.
- *Actions* are tasks that require no problem-solving. They are essentially stimulus-response activities.
- In addition to goals, tasks and actions, *devices* also play a part in achieving goals. A device is something which is used by humans to change the system to a desired state in order to achieve a goal.

4.5.1 Task Analysis Methods

There are several methods for analysing tasks which Dix (1997: XX) - in an effort to resolve some of the confusion surrounding task analysis - classifies on the basis of whether the methods focus on the user or on what the user does. He groups the methods into cognitive modelling methods, which are a way of modelling the user and the user's abilities and processes, and task analysis proper, which is concerned with analysing the actual tasks that users carry out.

Cognitive models such as *Goals, Operators, Methods and Selection Rules* (GOMS), *Cognitive Complexity Theory* (CCT) and *Knowledge Analysis of Tasks* (KAT) are characterised by the fine level of detail with which they can analyse tasks. This results in high levels of complexity and difficulty of use which are criticised by Landauer (1995:286) and Preece (1994:426). Although this complexity is not as problematic when analysing interactions with software (to which these methods are better suited), it presents significant problems in the context of this study. One possible goal of a user reading a user guide could be "to learn how to use the software". However, a goal of this broad nature is too high-level for a GOMS or CCT analysis and would rapidly become cripplingly complex. Furthermore, the reliance of KAT (see Johnson & Johnson 1991) on interviews and questionnaires calls the reliability of the data into question because there is frequently a discrepancy between what subjects do and what they **say** they do. Coupled with the large amounts of resources, ancillary skills and time needed to implement these models, these models are not suitable for the purposes of this study.

4.5.2 Hierarchical Task Analysis

It is clear from the preceding paragraphs that cognitive models are unsuitable for our purposes here. In any case, there is sufficient literature available on reading to render such detailed approaches unnecessary. There has been a trend in recent years to adopt less detailed approaches which incorporate more observable behaviours on the part of users (Preece 1994:426,429). These task-orientated methods are designed to observe users from the outside in terms of the actual tasks they perform (Dix 1998:262).

Hierarchical Task Analysis (HTA) is, perhaps, the most common type of task-orientated analysis. This method involves dividing high-level tasks into their constituent subtasks, operations and actions and presents a hierarchy of tasks in either textual or graphical form. Presenting the hierarchy in graphical format is favoured on account of its clarity and accessibility (Preece 1994:413; Dix 1998:264). There are various types of HTA (Preece *ibid.*) but we will restrict ourselves here to discussing what is generally regarded as HTA

In general terms, HTA involves identifying tasks using a variety of means such as interviews, direct observation, examining operating manuals and job descriptions etc. and then breaking these tasks down into subtasks. Subtasks are then divided into their constituent subtasks and so on. In breaking tasks into subtasks, it is necessary to decide on the level of detail or granularity needed to understand the tasks. It is conceivable that we could continue breaking tasks down into increasing numbers of subtasks *ad infinitum*. Curiously, there are few hard and fast guidelines regarding where the subdivision of tasks should stop. Much of the literature on this subject simply states that the main task should be broken down into between four and eight subtasks (Preece 1994:416; Nectar 1999:3). Apart from this there is little guidance other than to decide how detailed the analysis should be before starting the analysis and to ensure that all hierarchies are consistently detailed.

Dix (1997:264) maintains that some form of *stopping rule* is needed to decide when the analysis is sufficiently detailed. This, he maintains, will depend on the specific purpose of the task analysis. He proposes two possible stopping rules. The first is, he maintains, particularly suited for use in the design of training materials. Known as the *P x C* rule, this rule states that "if the probability of making a mistake in the task (P) multiplied by the cost of the mistake is below a threshold, then stop expanding". Thankfully, he elaborates upon this by saying that unless a simple task is critical to the overall task, it should not be expanded. This explanation notwithstanding, the problems in terms of calculating the probability and cost of errors would be prohibitively complex and difficult particularly as Dix gives no indication of how to produce values for P and C or for the threshold.

The other rule proposed by Dix is that where tasks contain complex motor responses or internal decision-making, the subdivision of tasks should stop. This applies except where the decision-making is related to external actions such as opening a manual.

Returning to the creation of a task hierarchy, simply having such a hierarchy alone is not enough to gain a useful insight into how a task is performed. With this in mind, HTA introduces the concepts of goals, operators and plans. If we consider goals as the desired system state or what it is users want to achieve, tasks describe the way a goal can be achieved while operators are the lowest level of behaviour (Preece 1994:413). However, there may be several ways of achieving a goal, where each method can involve multiple tasks. The critical factor here is how these methods are selected and in what order. This is where *plans* are included in the hierarchy. Plans specify the way subtasks are performed and the conditions under which each constituent subtask needs to be performed. In the task hierarchy, plans are numbered according to the tasks they relate to.

4.5.2.1 Conducting a Hierarchical Task Analysis

With regard to conducting a hierarchical task analysis, Preece (1994:416) splits the process into three stages:

Starting the Analysis

This involves specifying the main task to be analysed. This task is then broken down into the requisite number of subtasks. The subtasks should be specified in terms of objectives and together they should cover the entire area under study. Next, the subtasks should be represented as layered plans while ensuring that they are correct and complete.

Progressing the Analysis
This stage involves determining the level of detail to be achieved with the analysis. It is essential that all subtasks are treated consistently in this regard. Once all of the subtasks have been divided into their constituent task components, each subtask and plan is numbered using a hierarchical numbering system.

Finalising the Analysis
During this stage, the analysis is checked for consistency and completeness. It is recommended that the analysis be checked for omissions and consistency by someone who has not been involved in the task analysis but who has a thorough understanding of the task.

As stated previously, the analysis can be presented either textually or graphically. Ultimately, the decision to use one or other method is a matter of personal preference but there are a number of points to consider:

- textual representation may be more compact in terms of page space but it is less accessible than graphical representation (Dix 1998:264)
- graphical representation is clearer and more accessible but it can be considerably larger and less compact

4.5.2.2 *Task Analysis of Reading User Guides*

Before we can even contemplate conducting any form of task analysis on the way user guides are used, it is essential to understand **why** users read them. As we discussed in Chapter 2, the primary purpose of user guides is to educate users. However, users may have different learning strategies when it comes to software: some may perform tasks as they read the user guide while others may read the user guide completely before starting to use the software. Others may quickly browse the user guide to find key information before learning the rest as they use the software. These different circumstances are known as *scenarios of use* or *use cases* (UsabilityNet:g) and they specify how users carry out their task in a specific context.

There are several potential use cases for user guides. Coe (1996:138-140) provides a useful categorisation of these as part of an overall discussion of reading strategies. She identifies two primary types of reading strategy: *reading to learn* and *reading to perform actions*, corresponding to users' desire to acquire declarative knowledge or procedural knowledge respectively. Users may also access a text in a number of ways: they may read a text sequentially or randomly. In addition to the general purpose of reading expressed by the type of strategy and the methods of accessing the text, Coe (*ibid.*) maintains that there are five goals of reading. These are closely related to the type of reading and they describe the method of reading chosen by the user in order to achieve the desired outcome:

- **Skimming** involves reading for the "gist" of a text. This is a declarative goal and can be either sequential or random in nature.
- **Scanning** is another declarative method intended to find specific information.
- **Searching** is a declarative method which is similar to scanning but in this case the reader's attention is focussed on the meaning of specific information.
- **Receptive** refers to reading to fully understand information in a text. It can be either declarative or procedural and it can be either sequential or random.
- **Critical** refers to reading for evaluation purposes. A sequential method that is either declarative or procedural.

Assumptions about Reading User Guides

For the purpose of our study, we will presume that the use case involves users reading the text to perform some task, i.e. they are looking for procedural information. Novice users are, after all, more interested in how to do something (Redish 1988) rather than acquiring a deep theoretical understanding of the underlying principles. We will also assume that users will access information sequentially, at least within a section. Thus, while users may not read every section in sequence, they will at least proceed through each section in the intended order. This is because some readers may, for whatever reason, skip certain sections usually because they may already know some of the information presented. Finally, we will assume that the reading goal is receptive. Users want to fully understand the tasks they are performing.

Having placed the user guide and the user in a specific use case, we are in a position to begin our task analysis. Fortunately, a lot of information is already known about the processes involved in reading so we are not forced to start entirely from scratch. Coe (1996:134-135) provides a useful and reasonably detailed summary of the cognitive processes of reading which is based on the following headings:

1. Perceive visual data
2. Recognise words and letters or learn new words
3. Understand the relationship of words to the entire passage
4. Relate the information to a body of knowledge
5. Encode the information
6. Retrieve the information
7. Communicate the information

In the particular use case we have specified for this task analysis, only Stages 1 to 6 are directly applicable because we are using the information immediately in order to perform a task. We are not communicating this information to anyone else although Coe maintains that Stage 7 *can* involve communicating the information to oneself.

Concentrating, instead, on Stages 1 to 6, we can say that in **Stage 1**, perceiving information requires the physiological detection of physical stimuli. This information is then subjected to pre-attentive processing to group the physical marks on the page into groups which may have a meaning for us. In **Stage 2**, we take these shapes and group them into letters and words. Using a combination of pattern matching techniques (prototype matching, template matching, distinctive features) we match these words with lexical information in long-term memory (LTM). This process identifies the shapes as words the meanings of which we may or may not have in semantic memory. Where we do not have semantic information associated with the words, procedural memory is activated to provide us with a way of finding out what the unrecognised or new words mean. Such procedures might include how to use a dictionary etc. Once we have located the meaning of the word, we store it in LTM for later use or in short-term memory (STM) for immediate use.

Once we have recognised and identified all of the words, **Stage 3** involves relating these words to each other and with the rest of the sentence, paragraph or text. This requires the retrieval of semantic information for each word and the reconciliation of the various meanings for each one within the context of the meanings of other words. We then chunk this information and combine each chunk with additional chunks. Once we understand the text or passage, we then relate the information it contains to what we already know about the subject. In **Stage 4**, we may create new schemes or modify existing ones. This information is then incorporated into our existing knowledge base which is stored in LTM.

In **Stage 5**, we encode the information in order to integrate it into new or modified schemes. This takes place in STM and the information is encoded as either procedural or declarative information. Ultimately, it is stored in LTM with varying degrees of success (see page 84).

In **Stage 6**, we retrieve information from LTM. In our specified use case, this would be necessary if, at a certain point in reading the user guide, there was insufficient information available to perform a task. Consequently, the information is either maintained briefly in STM or placed in LTM until the remaining information which is needed becomes available.

These stages govern the cognitive processes of reading in general. There are, however, certain additional factors which distinguish reading in general from reading a user guide to do something. To understand this, consider the following. Once we have performed the processes mentioned above, we are in a position where we have information either in STM or in LTM. The question now arises as to how we are going to use this information. If we are reading to perform a task, our first problem is to decide whether we have sufficient information available to perform the task. As mentioned in Stage 6 above, we may not have all of the information necessary, in which case the information is stored in STM or LTM and we continue reading until we find the required information.

If we do have all of the information needed, we have to locate the information in LTM. Providing this is completed successfully, we then decide how to perform the task. The question arises as to whether the task can be performed (a) at once, or (b) do we need to divide it into subtasks. If the answer is (a), we perform the task, ensure that we have been successful and continue reading. If the answer is (b), we divide the task into subtasks, perform each one (if possible) and in sequence and then continue reading.

Appendix K illustrates a hierarchical task analysis of the process of reading a user guide. This task analysis uses the methodology and notation described on page 127. Admittedly, this is a simplified analysis in that certain processes either are not, or cannot, be represented, e.g. selection rules for multiple possible actions, while others are omitted to minimise repetition and to improve the clarity of the diagram. Furthermore, it does not take into account phonic knowledge, related word forms or systematic ambiguity. Nevertheless, the task analysis is useful in that it helps us visualise the major procedures and tasks involved in reading and using a user guide.

Findings of Task Analysis

The task analysis clearly shows the primary tasks involved in reading a user guide (*Tasks 1-6*). These tasks are governed by *Plan 0*, i.e. do tasks 1 to 6. *Task 1* is not expanded because the activities involved are basic physiological functions. *Task 2* consists of tasks 2.1-2.4. These tasks are governed by *Plan 2* which says that *Tasks 2.1, 2.2* and *2.3* should be carried out. If *Task 2.3* is successful, proceed to *Task 3*. If *Task 2*.3 is unsuccessful, do *Task 2.4*. The process proceeds through each primary task and its associated subtasks in a similar way.

It is clear that certain areas of the process are particularly taxing for readers in terms of the amount of cognitive processing required and the loads placed on their memory resources. For instance, *Task 3.3.1* requires a significant amount of cognitive effort to analyse information and retrieve information from long-term memory. This load can, for example, be reduced if procedures and their instructions are repeated enough times so that they become automated and require less cognitive effort (Gavin 1998:33; Raskin 2000:18,19). This reduces the need to analyse and process information in as much depth and speeds up retrieval of the actions needed to perform tasks. Other areas which require large amounts of processing include *Tasks 2-4* and *Tasks 6.1.1-6.1.3*, *6.1.5* and *6.1.5.1-6.1.5.2*. We can reduce the amount of processing required and shorten retrieval times using a variety of measures such as priming or providing introductory information to prepare readers for what is to come, e.g. stating objectives at start of chapter (cf. Foss *et al.* 1981).

Similarly, *Tasks 3.1* and *3.2* place serious burdens on readers which can be reduced by using clear, simple and familiar language and words.

The task analysis also shows us areas which place burdens on readers which may not necessarily be the focus of a technical communicator's activities. For example that we need to be able to recognise vast amounts of information relating to letters and words is clear from *Task 2* and its subtasks. However, this is inherent to any form of reading and does not necessarily represent an area where we can make significant changes. This method can also highlight the need for external resources such as dictionaries or reference materials. Of course this task analysis is restricted to dictionaries, but other external resources such as encyclopaedias, help files, technical support and so on could be included.

4.6 IMPLEMENTING THE DESIGN

In the preceding sections we examined various methods for analysing tasks and more importantly how these tasks are carried out. We also conducted a rudimentary task analysis on the processes of reading in general and specifically of using a user guide.

The purpose of task analysis is to show the context in which the interaction and tasks take place. In addition, task analysis shows us the potential weaknesses in the process and identifies areas where we can improve or streamline the usability of the system or interactions with it. However, task analysis does not and cannot tell us **how** we can achieve the goal of improving usability in the interaction nor can it tell us how to fix the problems contained therein.

Consequently, we need some way of connecting the knowledge of human cognition detailed in Chapter 3 with our knowledge of the interaction between the human and the system via the user guide as illustrated in the task analysis. This is achieved through the formulation and implementation of *principles*, *guidelines* and *rules* (Dumas & Redish 1999:53-61; Schneiderman 1998:52ff).

4.6.1 Principles, Guidelines and Rules of Usability

> An interface is humane if it is responsive to human needs and considerate of human frailties (Raskin 2000:6)

In order to ensure that an interface is both "humane" and usable, we need to take the various characteristics of the human cognitive system - as described in the preceding chapters - into account when implementing an interface design. Using principles, guidelines and rules provides a way of selecting key cognitive issues which are of relevance to cognetics and transforming them into practical and workable methods for improving interactions and ultimately the usability of the interface.

Dumas & Redish (1999:52-53) assert that many of the usability problems encountered in practice are dealt with in the wealth of information obtained through HCI research and usability testing. They make the point that HCI and documentation design both draw on the same body of knowledge albeit from slightly different angles. They claim that HCI is concerned with designing software to ease the interactions with users while documentation designers design documentation that "works for users" rather than documentation that just describes the system (*ibid.*). Experts from both disciplines ask similar questions such as "*How do I make sure users' abilities are catered for?*" etc. The mass of knowledge is "distilled into general principles and guidelines" (Dumas & Redish 1999:53; Preece 1996:488-491).

Defining a principle as "a very broad statement" that is usually based on research into how people learn and perform tasks, the authors provide an example which applies both to computers and to documentation (Dumas & Redish *ibid.*): be consistent in your choice of words, formats, graphics and procedures. This principle is based on the fact that we learn faster when what we see and do is consistent (see page 90).

4.6.1.1 *Examples of Principles*

Before we begin discussing concrete strategies for improving the usability of our interface, i.e. software user guides, we should first identify the predominant principles of HCI and usability derived from studies on technical communication, cognitive psychology and HCI as described in previous chapters.

In what he terms the "*Eight Golden Rules of Interface Design*", Schneiderman (1998:74-75) sets out a series of principles which play an important role in designing usable, user-friendly and effective interfaces. While there are numerous aspects of HCI that can be drawn upon in interface design, these rules serve as a concise and general overview of the areas that need attention:

- Strive for consistency, use similar prompts, commands, fonts, layout, situations, instructions etc.
- Enable frequent users to use shortcuts
- Offer informative feedback
- Organise sequences of actions so that they have a start, middle and end.
- Offer error prevention and simple error handling
- Permit easy reversal of actions
- Support the internal locus of control, this allows users to feel in charge of the computer and not vice versa.
- Reduce short-term memory load

It is apparent that the principles set out by leading figures in HCI such as Schneiderman share more than a few similarities with those produced for documentation design (see Section 2.3). Dumas & Redish (1999:61) explain that this similarity is due to the fact that the principles for both are based on creating products that are usable.

Due to the ubiquity of evaluation throughout the development and production process numerous evaluation criteria can also be used as design principles. One such set of criteria was developed by Nielsen (Molich & Nielsen 1990). His *heuristic* evaluation method was developed as a cheap evaluation tool for companies who "couldn't afford or hadn't the resources for empirical usability testing" (Hill 1995:119). Heuristics are general design principles that are usually, but not always, effective (Landauer 1995:283). The following paraphrased list of Nielsen's heuristics represents what are widely acknowledged to be best practice for ensuring usability.

- Use simple and natural language.
- Say only what is necessary.
- Present the information in a logical way.
- Speak the users' language - use familiar words and concepts.
- Minimise users' memory load.
- Be consistent.
- Provide feedback and tell users what is happening.
- Provide clearly marked exits to allow users escape from unintended or unwanted situations.
- Provide shortcuts for frequent actions and users.
- Provide clear, specific error messages.
- Where possible, prevent errors by limiting the number of available options or choices.
- Provide clear, complete help, instructions and documentation.

These principles are widely cited by other sources such as Landauer (1995:283), Hill (1995:119-120) and Dumas & Redish (1999:65).

4.6.1.2 Examples of Guidelines

Where principles are goals or ideals, they do not say how to achieve the goal. Guidelines are more specific goals and they explain how a principle can be implemented. Dumas & Redish (*ibid.*) state that any one principle can give rise to numerous guidelines although not all of them may be applicable at the same time. Thus, guidelines are derived from principles for a specific context and set of circumstances. Crucially, Dumas & Redish claim that HCI principles and guidelines are only intended to supplement usability testing and that there is no guarantee of a completely usable design, even if all of the relevant principles and guidelines are followed. However, adhering to guidelines makes the incidence of serious usability flaws less likely. Guidelines based on the aforementioned principles might include:

- Always phrase instructions consistently
- Avoid excessively long sentences
- Only use approved terminology
- Use the same formulations and constructions for sentences
- Avoid confusing verb tenses

4.6.1.3 Rules

Although guidelines are more explicit than principles, they are not explicit enough with regard to actually implementing principles. Thus, Dumas & Redish introduce the notion of "local rules" (1999:58). Local rules provide clear, unambiguous and repeatable strategies for implementing the knowledge provided by principles. For example, if we use the principle "be consistent" and develop a guideline for it like "use the same formulations and constructions for sentences", we could produce the following rules:

- Always use active verb constructions when describing actions performed by the system
- Only refer to the product as X, never as Y or Z.
- The verb "run" must be used instead of "execute" or "call".
- Conditional sentences must take the form "If [condition], then [action]"
- Sentences must not exceed 20 words

A significant difference between guidelines and local rules is that while guidelines may conflict with each other in certain situations, rules **always** apply; they are absolute constants in the context where they apply.

Creating and compiling sets of guidelines and rules is complex and time consuming according to Dumas & Redish (1999:60). In this regard, they say that it is not always necessary to do so because there are numerous sources of guidelines and rules available. Such guides include the Microsoft Style Guide, the SAP Guide to Writing English, AECMA and so on. These publications provide a way of implementing various principles and guidelines without the effort or expense of drawing up complete sets of rules for each context.

4.7 Iconic Linkage

With the knowledge gained from the previous sections, the next step is to establish how to improve the interface. Having discussed the concepts of principles, guidelines and rules it is now time to implement our understanding of human cognition, usability and cognetics. We can do this by selecting one guideline and implementing it in a practical context in order to gauge its overall effect on usability. The following sections introduce Iconic Linkage as one possible guideline and discuss its associated rules while seeking to elaborate on its practical implementation. The potential benefits of implementing this guideline will also be discussed.

4.7.1 What is Iconic Linkage

Iconic Linkage (IL) refers to the use of isomorphic constructions to express what is essentially the same information. So, where the same information is expressed more than once in a text, the exact same textual formulation or construction is used. This is in contrast to the use of slightly different formulations which is employed in order to prevent what is commonly seen as repetition.

In the case of translated texts, IL is the repetition or re-use of target language translations for source language sentences which have the same meaning but different surface properties. In other words, sentences which are semantically identical but which are non-isomorphic are translated using the same target language sentence or construction.

4.7.1.1 Origins of Iconic Linkage

The term Iconic Linkage was coined by House (1981:55) to refer to instances of structural similarity in adjacent sentences. She defines IL as occurring when two or more adjacent sentences in a text "cohere because they are, at the surface level, isomorphic". This phenomenon is quite similar to what technical writers call "parallelism" (see page 41). Parallelism is a phenomenon which is widely recognised as a desirable feature of sentence structure (D'Agenais & Carruthers 1985:104; Mancuso 1990:231; White 1996:182). Essentially, parallelism means that parts of a sentence which are similar, or parallel, in meaning should be parallel in structure.

Parallel constructions can also be described as instances where two or more groups of words share the same pattern (White 1996:182). Thus, we can see that parallelism can occur on both a sentence level and on a sub-sentence level. The following sentences 1a and 1b illustrate parallelism.

1a: If you want to open a file, click Open.

1b: If you want to close a file, click Close.

Table 6: Example of Parallelism Between Two Sentences

When there is a lack of parallelism (for example in 2a and 2b) some of the grammatical elements in a sentence do not balance with the other elements in the sentence or another sentence. Consequently, the clarity and readability of a section of text are adversely affected. What makes this undesirable, apart from potential grammatical errors, is that it distracts the reader and prevents the message from being read quickly and clearly (Mancuso 1990:232).

2a: If you want to open a file, click Open.

2b: The Close button must be pressed when you want to close a file.

Table 7: Example of Two Sentences which do not have Parallel Structures

Parallelism is not just important in avoiding grammatical and comprehension problems, it is also very useful in reinforcing ideas and learning. The grammatical symmetry of parallelisms helps readers remember information more easily (White 1996:183).

Where my definition of IL differs from both House's definition and parallelism is that parallelism and House's definition deal with localised instances of structural similarity. Both deal with isolated pieces of text at particular locations, e.g. a sentence or list. Instead, IL as used here is an active strategy which is used throughout a text. Indeed, instances of IL can be separated by large stretches of text. In addition, rather than being restricted to individual phrases or sentences, IL can manifest itself in entire paragraphs or longer stretches of text.

In contrast to House's definition, IL is actively introduced into a text, rather than being a naturally occurring feature of a text, i.e. a feature of the text when it was **first** produced.

4.7.2 Latent and Introduced Iconic Linkage

Before embarking on a more detailed discussion of Iconic Linkage, it is important to differentiate between the two principal types of Iconic Linkage: *Latent* and *Introduced.*

Latent Iconic Linkage refers to isomorphic, semantically identical sentences which occur "naturally" in a source text. These instances of Iconic Linkage form part of the text as it was originally written. Frequently, such instances of Iconic Linkage are destroyed during subsequent processes such as editing or translation. This can occur for a variety of reasons. With regard to the translation process, translators may not always remember how they dealt with a previous instance of a sentence and will provide a slightly different rendering. While this problem may be lessened though the use of computer-aided translation (CAT) tools such as Translator's Workbench or Déjà vu, not all translators have access to such tools. Another reason for the loss of latent Iconic Linkage both during translation and during editing is that repetition can be regarded as anathema to good writing. While this may be the case with creative and other forms of writing, repetition (or consistency) is actually desirable in technical documents in general and user guides specifically. As such, all instances of latent Iconic Linkage represent naturally occurring "quality components" within the text and should be maintained during translation and editing.

Introduced Iconic Linkage refers to instances of Iconic Linkage which are added to a text *during re-writing, editing or translation.* If for example, two or more non-isomorphic but semantically identical sentences (or parts of sentences) are encountered by an editor or translator, using the same target text construction to render each source sentence will introduce Iconic Linkage into the text.

4.7.3 Types of Iconic Linkage

There are two fundamental types of Iconic Linkage: *Full and Partial.* Each type is determined by the extent of repetition within a particular sentence. The following paragraphs describe the nature of each type.

4.7.3.1 *Full Iconic Linkage*

Full Iconic Linkage refers to complete, discrete and relatively independent units of text such as sentences, headings, table cells etc. which are semantically identical and which are also isomorphic. Rather than writing the same information using different constructions or formulations, the same unit is repeated at different points in the text. Table 8 illustrates examples of full Iconic Linkage.

Matching Paragraphs

Full Iconic Linkage, i.e. identical sentences may be combined to form larger sections of isomorphic, semantically identical text. Thus, several instances of full Iconic Linkage can form iconically linked paragraphs which can then in turn be combined to form even longer stretches of iconically linked text. This is illustrated in Table 9.

4.7.3.2 *Partial Iconic Linkage*

Partial Iconic Linkage refers to *parts* of a unit that are identical – this is usually because there are certain factual differences which mean the units are not complete semantic matches. It can also be because one unit communicates more information than the other. Ideally this should not happen because best practice in technical writing holds that a sentence should only communicate one idea at a time. This also complies with the principles of good design set out by cognitive design principles to reduce STM load (see Chapter 3). Table 10 illustrates some examples of partial Iconic Linkage.

4.7.4 Examples of Latent Iconic Linkage

The following tables illustrate the various forms of latent Iconic Linkage: *Full Iconic Linkage*, *Matching Paragraphs* and *Partial Iconic Linkage.*

Examples of Full Iconic Linkage

Source	Pages	Sentence
Cylon UCC4 Keypad UG	5-7	If you make an error, press the **Clear** key.
Cylon UCC4 Keypad UG	6, 7, 9, 12, 13, 14	The menus shown below are examples only and may not appear on your keypad.
Cylon Engineering Tool	19,20	The **Save As** dialog box will appear.
Cylon Engineering Tool	88-90	Use another universal controller.
Cylon UCC4 Programming	34,39	Present Value This is the present value of the global
Informix TP/XA	2-9, 2-10	The following table describes the XA-related flags.

Table 8: Examples of Latent Iconic Linkage – Full

Examples of Matching Paragraphs

Source	Page	Sentence
Cylon Engineering Tool *	61,62	No. of Modules Services The value entered here dictates how many modules are to be serviced. This generally equates to the total number of modules contained in the strategy. 0 in this box indicates that the UC16 is not servicing the strategy and the green LED pulses on and off.
Cylon Engineering Tool *	61,62	When Composed This is the time and date that the strategy was commissioned.
Cylon Engineering Tool *	62,63	User ID This shows who commissioned the strategy.
Cylon Engineering Tool	79,84	1. Go "On-Line" by logging into the desired UC16 To ensure that it is possible for the PC to communicate with the desired UC16, it is necessary to log in to that UC16 and get some information from it, e.g. its ROM version number, or its time. There are a number of ways in which a UC16 can communicate with a PC: Via a direct RS232 link between the PC and the maintenance port on the UC16 Via a UCC4 connected to an ARCNET with an RS232 link to a PC Via RS485 link to a UCC4 connected to a PC via RS485 maintenance port
Cylon UCC4 Programming	33,38	Enter the number of the global and click on OK. The LOCAL GLOBAL dialog box will appear.
Cylon UCC4 Programming	34,39	The source depends on the point type that was selected. If the point type is a Time Schedule then the number entered here is a time schedule number. Otherwise the number entered here is the UC16 number. Enter the time schedule or point number.
Cylon UC16 Programming	122-125/6	A datalog on an input point can be used to monitor how the plant is behaving with regard to such things as temperature and pressures. A datalog on an output point can be used to do such things as monitor valve positions. The pulse logs can be used to monitor items like energy usage and flow meters.

Table 9: Examples of Latent Iconic Linkage - Matching Paragraphs

In Table 9, the paragraphs marked with * follow each other, so as well as being matching paragraphs, the represent matching stretches of text.

Examples of Partial Iconic Linkage

Source	Pages	Sentence 1	Sentence 2
Cylon UCC4 Keypad UG	12-13	You can only *view* the date and time if your keypad has been programmed to do so.	You can only *change* the date and time if your keypad has been programmed to do so.
Cylon UCC4 Keypad UG	7, 9, 12, 13, 14, 17,	You can only *XXXX* if your keypad has been programmed to do so. (x6)	
Cylon Engineering Tool	19	If the *globals have* been saved as a UCC4 file before, select **Save UCC4** from the **File** menu.	If the *file has* been saved as a UC16 file before, select **Save UC16** from the **File** menu.
Cylon Engineering Tool	19	This will save *them in a file* automatically with the extension **.cmn.**	This will save *the file* automatically with the extension **.sta.**
Cylon Engineering Tool	28	*ON* This is the time the Schedule *starts*.	*OFF* This is the time the Schedule *ends*.
Cylon UC16 Programmin g	122,125	Up to *8* datalogs may be stored by a *UC12*.	Up to *16* datalogs may be stored by a *UC16PG or UC16IP*.
Lexmark UG	20, 21, 23, 25, 26	Load up to *XXXX* vertically against the right side of the automatic feeder. (x5)	
Digitech RP7 UG	14,15	Speed: Controls the speed of the *XXX* sweep. Ranges from 0 to 100. (x3)	

Table 10: Examples of Latent Iconic Linkage – Partial

4.7.5 Examples of Introduced Iconic Linkage

As stated in Section 4.7.2 above, Iconic Linkage can be introduced into a text either during the initial writing process or during editing, re-writing or translation. The following examples, based on the user guide for a software package called *DigiLog* (see Section 5.4), illustrate the introduction of Iconic Linkage in a monolingual context, i.e. production or editing. These examples present sentences which are all semantically identical. Each sentence in each group can be used to replace the other two sentences in each group.

Examples of Introduced Full Iconic Linkage

1a	Enable *QuicKeys* by clicking *Use QuicKeys*.
1b	*QuicKeys* are enabled by clicking *Use QuicKeys*.
1c	To enable *QuicKeys*, click *Use Quickeys*.
2a	If no checkmark appears before the *Use QuicKeys* option, the *QuicKey* function is disabled.
2b	If there is no checkmark in front of the *Use QuicKeys* option, the *QuicKey* function is disabled.
2c	*QuicKeys* are disabled when there is no checkmark in front of the *Use QuicKeys* option.
3a	Select the *Exit* option to close the *DigiLog* program.
3b	Click *Exit* to terminate *DigiLog*.
3c	To close *DigiLog*, click *Exit*.

Table 11: Introduced Iconic Linkage in Monolingual Context

The sentences shown in Table 12 (from the author's personal translation archive) provide examples of Iconic Linkage introduced during translation and are taken from a user guide for a food packaging machine and its accompanying software. They illustrate how non-isomorphic but semantically identical sentences in a source text can be rendered with Iconic Linkage in the target text. The various sentences have been analysed using the Trados Translator's Workbench CAT tool to illustrate the apparent match between sentences based on surface characteristics.

Original Instance	Wechsel in den Programmier-Modus (PROG-Modus).
Second Instance	Wechseln Sie in den PROG-Modus.
Translation	Switch the machine into program mode (PROG Mode).
(The two sentences are a 54% surface match but actually mean the same thing)	
Original Instance	Maschine ist im Arbeits-Modus.
Second Instance	Die Maschine befindet sich im Arbeits-Modus.
Translation	The machine is now in Work Mode.
(The two sentences are a 67% surface match but actually mean the same thing)	
Original Instance	Wollen Sie ohne Begasung arbeiten, stellen Sie den Parameter-Wert Gas/Restvakuum auf den selben Wert ein, wie den Parameter-Wert Vakuum.
Second Instance	Für das Arbeiten ohne Begasung müssen Sie den Parameter-Wert Gas/Restvakuum auf den gleichen Wert einstellen wie den Parameter-Wert Vakuum.
Translation	If you want to work with gassing activated, use the same value for Gas/Residual Vacuum and Vacuum.
(The two sentences are a 72% surface match but actually mean the same thing)	

Table 12: Examples of Introduced Iconic Linkage – Full

Examples of Introduced Partial Iconic Linkage

The sentences shown in Table 13 provide examples of introduced partial iconic linkage. These examples differ from each other in terms of certain material facts, e.g. in each case a different button must be pressed or a different system state occurs.

Original Instance	Drücken Sie die PROG-Taste.
Second Instance	Drücken Sie die ENTER-Taste.
Translation	Press the XXXX button.
(The two sentences are a 96% surface match)	
Original Instance	Die Maschine bleibt im PROG-Modus.
Second Instance	Die Maschine bleibt im *Arbeits*-Modus.
Translation	The machine is now in *XXXX* Mode.
(The two sentences are a 67% surface match)	

Table 13: Examples of Introduced Iconic Linkage – Partial

4.7.6 Iconic Linkage as a Cognitive Strategy

As we discussed in Section 4.6.1, principles are general goals or objectives which may give rise to several guidelines and local rules. So for example, the principle which states that a user's STM capacity of 7±2 chunks of information should not be overburdened might give rise to guidelines to the effect that sentences should be kept short and contain simple and familiar words (which can be grouped into larger chunks of information, thereby reducing the number of chunks the reader has to retain).

However, these guidelines can also prove effective in implementing other, perhaps completely different principles. For instance, the idea of using simple and familiar words may also be used as a way of minimising users' memory load because people find it easier to recall the meaning of words which are used frequently. Similarly, while keeping sentences short may reduce STM load, it can also reduce the possibility of the reader becoming distracted by some other stimulus. This idea is related to the principles of saying only what is necessary, presenting information in a logical way, and preventing errors by limiting the number of available options or choices (see page 138). As Kellogg puts it

> ...adopting cognitive strategies that circumvent attentional overload may be one key to optimum writing performance (Kellogg 1988:355-6)

If we relate Iconic Linkage to the concept of principles, guidelines and rules, we can say that Iconic Linkage is a guideline which states "always present information which is repeated throughout a text using exactly the same formulation of words and structures". This guideline itself draws on several principles such as consistency, reducing STM load etc. (see *Memory* on page 67) but gives rise to manifold rules which can consist of various efforts to regulate the ways in which information can be formulated and presented in a text.

4.7.6.1 *The Benefits of Iconic Linkage*

The idea of implementing Iconic Linkage in a user guide draws on several areas of cognitive psychology and represents a way of implementing a variety of principles. On the basis of the areas examined in Chapters 3 and 4 we can see that, in theory at least, Iconic Linkage presents the following benefits in the production of user guides:

Reduction of Short Term Memory Load

Iconic linkage reduces the demands placed on short-term memory by helping to chunk information into meaningful and manageable units. Not only does this help maximise on STM capacity but it also reduces the amount of processing needed and speeds up the retrieval of information (see page 70).

Habit Formation

Iconic Linkage facilitates the formation of good habits. Raskin (2000:20) maintains that one obstacle to the formation of habits is the provision of multiple methods of doing something because it shifts the attention away from the task to the process of choosing the method. By ensuring that information is phrased in a standard, uniform way, we eliminate the need for readers to decode the same information all over again. Thus, users concentrate more on the task at hand and not on the interface (user guide).

Subconscious Learning & Perceptual Learning

In a way similar to habit formation, the repetition of information phrased in precisely the same way takes advantage of latent learning (see page 89) whereby users learn information without realising it or intending to.

As was discussed on page 73, perceptual learning is the way we know **how** to perceive sensory information each time we encounter it. This is important for Iconic Linkage in that we can change the response of users to sensory perception. For example, where the sight of a particular piece of text may result in confusion, over time it will provide reassurance, information etc. for readers.

Accessibility of Information

By consistently using the same constructions to present information, the text assumes a certain visual or iconic consistency with each visual representation being associated with the information it conveys in image memory. This in turn allows Iconic Linkage to take advantage of image memory (see page 71; see also Faulkner 1998:4) which reduces the processing load and speeds up retrieval of information by using recognition as a precursor to recall; recognition allows us to determine which information we have already seen and need to recall and which information we have yet to learn.

Retention of Information

Since repetition aids habit formation (see Section 3.5.3.5 on page 90), Iconic Linkage, by virtue of the fact that it repeats textual constructions, increases retention of information. Furthermore, borrowing from the theory behind parallelisms, the grammatical symmetry of parallelisms also helps readers remember information more easily and it reduces confusion while improving clarity and readability (cf. Mancuso 1990:231; White 1996:182).

Problem-Solving

Since Iconic Linkage reduces the cognitive resources needed to decode and access information, it reduces the amount of problem-solving required to comprehend a piece of text (see Section 3.5.3.6). This is because the "problem", i.e. a sentence, has already been solved and so each time the sentence is encountered the solution is reinforced rather than created again.

Improvement of Consistency & Predictability

It is clear that using the same phrases and constructions to present the same information improves the consistency and predictability of the text (see page 112). With no major surprises lying in wait for users, they can proceed through the text in relative comfort.

Reduction of Interference

By making texts more predictable and consistent, Iconic Linkage reduces interference between tasks. When we perform two tasks simultaneously, e.g. learning new information and using a piece of software, our performance in both will suffer, regardless of how automated one task (such as reading) has become. The more predictable, automatic and unconscious a task becomes, the less likely it will degrade or compete with other tasks for resources (see page 83). And so, rather than having the process of reading the user guide competing with the task of using the software for cognitive resources, we can automate the reading process even more. As a result we can reduce the cognitive demands reading makes on the user and free up more resources for the process of using the software.

4.8 CONCLUSIONS

Building upon the insight into cognition provided in Chapter 3, this chapter examined usability from the point of view of interfaces. In the context of human-computer interaction, an interface can be defined as anything that acts as an intermediary between a user and the inner workings and processes of a computer or piece of software. Having described usability in Chapter 3 as ensuring that an interface takes into account the cognitive abilities and limitations of humans, this chapter turned the notion of usability into a series of definite design goals and objectives which ensure users can work effectively, efficiently and with ease. These goals codify those aspects of human cognition where active strategies can be implemented to improve the interaction.

Cognetics, or cognitive engineering, was introduced as a discipline where interfaces are engineered or designed to capitalise on human abilities and to compensate for human limitations. There are several approaches to engineering interfaces and hierarchical task analysis (HTA) was chosen because of its ease of use, clarity and flexibility. An analysis was carried out of precise areas of the user guide interface where we need to concentrate our efforts to improve usability. The other methods mentioned, which are intended for use on software, would be excessively complex, if not impossible to apply to the cognitive requirements of a user guide.

With the knowledge gained from this analysis, the next step was to establish how to actually improve the interface. The chapter then discussed the concepts of principles, guidelines and rules and provided examples of each of these concepts.

Iconic Linkage (IL) was then introduced as a possible guideline for improving usability. We saw that IL can occur naturally within a text (i.e., when the text is first produced) or it can be actively introduced into a text. This can take place during initial production of the text, during editing or even during translation.

We have also looked at the extent to which partial of full IL can occur between two or more sentences – partial or full. Where only parts of the information payload in sentences are matched and this information is phrased identically, the IL is said to be *partial*. Where entire sentences are semantically identical, both can be written using the same structures and formulations. Such sentences are semantic matches and represent examples of *full Iconic Linkage*. Where several instances of full IL occur consecutively, they can represent *matching paragraphs* where the whole paragraph presents the same information in exactly the same way as in another part of the text. Again, several matching paragraphs occurring in succession result in larger sections of Iconic Linkage. As a guideline, IL represents a bridge between the various principles outlined in preceding chapters and the numerous technical writing rules.

The chapter concludes by reiterating the principles which can be implemented using IL. In the following chapter, we will set out, firstly, to implement IL in a real user guide and secondly, to establish whether it has a measurable effect on usability.

Chapter 5
EMPIRICAL STUDY

5.1 INTRODUCTION

The preceding chapters examined software user guides with a view to improving their usability. Iconic Linkage was proposed as one method of improving the usability of user guides. However, to be in a position to say with confidence that the presence of Iconic Linkage in a text does actually improve document usability, it is essential that this hypothesis be tested in the form of an empirical study.

This chapter deals primarily with an empirical study to test the effect of Iconic Linkage in a software user guide under conditions representative of real-world usage. A crucial part of such an undertaking is examining the various methods and procedures commonly used in studies of this nature. This chapter begins by examining a range of data collection methods used in usability studies and proceeds to select those which are compatible with the aims of this study. To gain a better insight into usability testing involving users, a review of previous usability studies is also conducted.

We then proceed to develop a model for conducting the empirical experiments including hypothesis, evaluation criteria, methods, materials and data analysis models. To conclude, this chapter will discuss and evaluate the findings of the empirical study.

5.2 USABILITY EVALUATION

Regardless of the time and effort spent on engineering usability into an interface using the variety of methods outlined in previous chapters, the only true way of establishing whether the interface is indeed usable is to conduct some form of usability evaluation. There are essentially two types of evaluation – formative evaluation and summative evaluation. The type of evaluation used depends on when the evaluation is to be carried out as well as what it is hoped the evaluation will achieve.

Formative evaluation takes place during the development process in order to detect potential problems before the design is actually implemented. The aim here is to improve the usability of the final interface (Preece 1993:108; Preece 1994:603).

Summative evaluation, in contrast, is carried out on the finished interface. The aim here is to determine the level of usability of the interface so that judgements can be made as to the overall usability and quality of the interface (Preece 1994:103). Summative evaluation is used to ensure that "the final product is up to standard, not to shape the design and development processes" (Landauer 1997:204).

For the purposes of this research, we are interested in assessing the overall level of usability of the final product, i.e. the user guide. For this reason, we will restrict ourselves to an examination of usability evaluation from a summative point of view.

5.2.1 Empirical Approaches to Evaluation

In conducting usability evaluations, there is a wide variety of approaches and associated methods which can be used. We can group these methods into two broad categories: *analytical* and *empirical* (Faulkner 1998:113).

Analytical evaluation, according to Preece (1993:109) uses formal or semi-formal methods of describing the interface in order to predict user performance. These methods include such strategies as GOMS and KAT which are described in Section 4.5. Analytical evaluation is primarily a formative approach and as such is of little significance here.

Empirical evaluation as described by Faulkner (*ibid.*) includes expert evaluation or heuristic analysis, observational evaluation, survey evaluation and cognitive walk-throughs (Hill 1995:120). Within this broad category of empirical evaluation, we can distinguish between *absolute* and *comparative* studies. Absolute experiments (for example, expert or heuristic evaluation) involve assessing the interface or system on the basis of predefined specifications, criteria and scores. Such standards might include, for example, Nielsen's heuristics (Nielsen & Molich 1990). Comparative experiments, on the other hand, involve assessing an interface and comparing it with some other alternative interface or version of the interface (Downton 1991:331; Faulkner 1998:113). For the purposes of this research, the comparative approach is preferable because we are concerned with determining whether Iconic Linkage can improve the usability of user guides in comparison to user guides where Iconic Linkage is not present.

We can further divide empirical evaluation into methods which involve users and methods which do not involve users. This distinction is important because Landauer (1995:281) maintains that, in contrast to analytical evaluation, expert evaluation and cognitive walkthroughs, testing involving users is the best way of evaluating usability. He says "only by studying real workers doing real jobs in real environments can we be sure that what we find out is truly relevant" (*ibid.*). With this in mind, the following section will discuss only those empirical evaluation techniques which involve users, namely observational evaluation, survey evaluation and experimental evaluation.

5.2.2 User Testing

A critical factor in any form of evaluation is the type of information the evaluation is supposed to provide: quantitative or qualitative. Quantitative data is numeric and is analysed using a range of statistical and mathematical methods. This, in some regards, makes it easier to process large amounts of data in order to provide statistical evidence. However, unlike qualitative data, it does not provide the detailed subjective data or opinions that give an insight into how users actually perform tasks using the interface. Qualitative evaluation does yield this information although the resulting data is more difficult to analyse in that it consists of a wide variety of diverse information which is often expressed in very different ways by users.

However, in the case of this study, we are interested in examining the way users work with the interface in a significant amount of detail. As we discussed in Chapter 3, usability is determined not only by how quickly or efficiently users work with an interface but also by users' opinions, satisfaction and attitudes to the interface. In addition to quantitative data such as how quickly users perform tasks, we are interested in finding out how easy users perceive the use of the interface to be. In addition, the decision as to whether to gather quantitative or qualitative data has significant implications for the design of the experiment, the methods used and the number of subjects involved in the experiment.

In the following sections, we will examine the various methods for conducting usability evaluations with users in order to gather a combination of both qualitative and quantitative data.

5.2.2.1 Observational Evaluation

Observing users performing tasks they would normally perform as part of their work is one of the most useful ways of collecting data about what users do when they use an interface and how they react to the interface. Users can be observed in a specially built usability lab (Preece 1993:112; Schneiderman 1998:128-130) or in the normal environment where the users work. The latter scenario is often referred to as a field study (Preece 1994:602). Observation can take a variety of forms and there are several methods for recording the information. There are two fundamental forms of observation: *direct observation* and *indirect observation*.

Direct Observation

Direct observation involves users performing tasks while an observer is present in the same room. The observer watches the user and makes notes, times actions or performs some other function such as asking questions etc. While this approach is valued for its informality and immediacy, there are a number of problems associated with direct observation. The first and perhaps most important issue with regard to the validity of the test results is that the presence of an observer can have a significant impact on the users' performance. This may be attributed to users believing that their performance is under constant scrutiny and that they need to "impress" the observer. The overall result is that users may actually perform better under test conditions than they would under normal working conditions, simply because they are trying much harder, a phenomenon known as the *Hawthorne effect* (Preece 1994:617; Faulkner 1998:122).

A second problem with direct observation is that the quality of information gathered and indeed the completeness of the information relies on the ability of the observer to correctly interpret what is happening and then to write accurate and useful notes. Furthermore, if the observer misses something, there is no way of capturing the lost data or referring back to what happened – the experiments are single-pass occurrences and the only record of events is the notes taken by the observer. Of course, using more than one observer may yield more comprehensive and complete data and also even counteract any possible biases associated with a single observer. However, if the presence of just one observer can affect a user's performance and distort the results of the test – the effects of several watchful observers could have disastrous consequences for the validity of the experiment.

Both of these problems, i.e. the Hawthorne effect and the unreliability/incompleteness of notes, can be counteracted through the use of indirect observation techniques (Faulkner 1998:122).

Indirect Observation

In contrast to direct observation where users perform tasks with an observer present in the same room, indirect observation involves users performing tasks without the presence of an observer. This type of observation generally incorporates some form of recording mechanism, be it in the form of audio, video or software recording or some combination of the three.

AUDIO RECORDING

Audio recording can be useful when combined with verbal protocols (see page 162 below). It involves recording what a user says during the course of an experiment (from general comments or spontaneous outbursts to verbalised thought processes in the case of think-aloud protocols). Audio recording and TAP are generally of greatest use in formative evaluations as the wealth of qualitative information can provide significant insight into the way the users interact with and perceive the interface. Particularly in the case of evaluations involving interactions with software, it can be difficult to match the audio recording with field notes on the events as there are no clues other than what users say and they may not always verbalise problem areas. It can be argued that TAP can place additional cognitive loads on subjects and can interfere with the way they perform tasks (see the discussion of attention in Section 3.5.2).

VIDEO RECORDING

Video recording or logging counteracts both the problems of direct observation and also the problems of audio recording in that it does not require the observer to be present in the same room as the user and it provides a permanent record of the experiment while allowing the observer to see what the user did at a given point in the experiment. It can also be used in conjunction with some form of verbal protocol (see page 162). By positioning cameras in a variety of locations in the room where the test is being conducted it is possible to capture a wide range of data such as what the user types on the keyboard, what appears on the screen, whether the user refers to the user guide, as well as the user's body language or facial expressions. With modern video camera technology, it is also possible to record reasonably high quality audio as well, thereby negating the need to synchronise different types of data (Dix 1998:429).

There are, however, certain problems associated with video logging. While this method provides valuable and comprehensive information, it only does this for as long as the user stays within the camera's field of view (Dix 1998:428). An obvious solution would be to position the camera closer to the user but then we risk undoing the benefits of indirect observation with the obtrusiveness of the camera. Conversely, hiding the camera and filming users surreptitiously raises certain ethical and legal issues and is to be avoided (Faulkner 1998:123; Dumas & Redish 1993:206). A simpler solution is presented by Dumas & Redish (1993:225). Rather than hoping that users stay in the correct position during the test, Dumas & Redish propose placing pieces of adhesive tape in an L-shape on the desk to indicate where the documentation must be placed. In addition to ensuring that the document stays within shot (if necessary) this strategy also constrains the users' movements and ensures that they stay in more or less the same position.

If recording events on the screen, careful positioning of cameras, the user, the screen and lighting is necessary unless there is some mechanism for connecting a second monitor (Dumas & Redish 1993:224, 384). In such a scenario, the second monitor could be placed in another room where the observer records the images on video. But the issue of data synchronisation can be quite problematic where more than one camera is used (Dix 1998:428; Preece 1994:619). At the very least some form of on-screen time code along with a mechanism for starting all of the cameras simultaneously will be necessary. This problem can be alleviated at least partially by through the use of software logging.

SOFTWARE LOGGING

Software logging is where the computer system records the user's actions during the test. There are two basic forms of software logging: one which records *time-stamped keypresses* and one which records the user's *interactions* with the system.

Time-stamped keypress logging records each key a user presses along with the time it was pressed. Certain varieties of keypress loggers also record system responses (Preece 1994:627) which means that error messages and dialog boxes can also be recorded.

Interaction logging tools operate in a similar manner except that they record the entire interaction in real-time. What makes interaction logging truly useful is that it allows the interaction to be replayed in real-time. This can illustrate additional information such as hesitations, visibly confused mouse movements and aborted attempts at tasks which might not be detected by keypress tools.

Apart from eliminating the need for a video camera to be recording the screen, the advantages of software logging are the fact that it is unobtrusive (although the same ethical questions apply as for video recording), it is at least partially automated and it provides a permanent record of the test. When used in conjunction with a video camera, it provides a comprehensive picture of the experiment from the point of view of the user, the interface and the interaction (Downton 1991:333). The only major drawback with this method is that it may require large amounts of storage space to deal with the frequently huge volumes of data and there may be synchronisation issues between the screen recording and the video recording (Preece 1994:266-267; Preece 1993:113; Dix 1998:428). However, this is offset by the fact that there is no need to analyse tapes from a second camera aimed at the screen.

INTERACTIVE OBSERVATION

Interactive observation is a type of indirect observation where the part of the system or computer is played by a member of the evaluation team. Commonly known as the "Wizard of Oz" (Faulkner 1998:122), this approach makes users think that they are using a real system, but in reality all of the system's responses and actions are performed by a human. This method is effective in that it does not require a fully functional version of the system and it can be implemented reasonably cheaply in comparison to the expense of producing a fully-functioning system. However, this approach is formative and is more suited to industrial situations where a system is actually being produced. The effort required to create an interface, in terms of design and labour coupled with the additional staff requirements to conduct the experiment make this approach difficult, if not impractical, in the context of this research.

Verbal Protocols

Verbal protocols are spoken records of users' comments, observations, exclamations and other information which may arise during the course of an experiment. One particular variety of verbal protocol is the think-aloud protocol which involves users saying what they are thinking, feeling, planning etc. as they perform tasks and use the interface. This can provide a valuable insight into what users want to do, what they think they are doing and their responses when something unexpected occurs. It is also possible to gain an insight into how users remember commands, plan and execute tasks and how they recover from errors (Preece 1993:113). Verbal protocols of this type are generally used in conjunction with audio or video recording (Preece 1994:621).

Although think-aloud protocols (TAP) are ideal for formative usability testing where the wealth of qualitative data they can provide is extremely useful in understanding the nature of the interaction, in the case of purely *summative* evaluations where, for example, the speed at which users work is being measured, TAP is less applicable, chiefly because summative evaluations require quantitative data. It can also be argued that TAP may hinder evaluation rather than aid it. This can be attributed to a number of factors. Firstly, as we discussed in Chapter 3 the human cognitive system can realistically deal only with one response to a stimulus at a time even though it can process several inputs or stimuli. Indeed, the process of dividing one's attention equally between two tasks is unreliable at best, but extremely difficult when performing two complex tasks such as problem-solving and verbalising thoughts where high levels of accuracy are required for both. Similarly, it is also held that the very act of putting into words what it is a user is doing will affect the way the user performs the task (Dix 1998:427; Downton 1991:334). While many agree that this double-tasking will degrade performance on both tasks (see page 83), there is some conflicting evidence that the think-aloud protocol may actually improve performance of the task. If this were proven to be true, it could be because the verbalisation process focuses a user's mind on the task and helps users rationalise the task better. Nevertheless, the additional strain of performing two complex and demanding tasks such as putting thoughts into words can result in lower performance and some users will simply be unable to verbalise their thoughts (Preece 1994:622).

There is also the problem of silence caused by the fact that users are either unaccustomed to thinking out loud or because all of their concentration is being devoted to performing the task; some users may simply forget to speak. This problem is discussed by Dumas & Redish (1993:278-281) who point out that while some people have no problems whatsoever in producing an "unedited stream of consciousness", others either mumble or do not speak. The authors make the interesting point that users need to be taught how to think out loud and that they may need to be reminded to do so. This in itself can become a source of stress for users who may already feel pressurised as a result of the tasks (Preece 1994:622).

In view of these problems, *retrospective* or *post-event protocols* are sometimes used to elicit verbal data from users. Instead of commenting on their actions while they perform them, users are shown a video of the experiment and are asked to comment on their activities. This approach produces different results in terms of the type of information users provide. According to Preece, users tend to rationalise or interpret their actions or even justify them (Preece 1994:623). Rather than simply stating what they were doing or thinking, users tend to explain what they are doing and why.

Dumas & Redish, however, do make the point that retrospective protocols frequently yield more suggestions as to how to improve the interface as compared to think-aloud protocols (Dumas & Redish 1993:279). However, we are not interested in using the evaluation to improve quality, merely to assess it.

Conclusions on Observational Methods

It is clear from the preceding paragraphs that observational methods are extremely useful in gathering comprehensive information on the way users work with an interface. While these methods produce large volumes of data which can be more difficult to analyse, the sheer detail and insight they provide more than compensates for this (Preece 1993:119). By using indirect observation we avoid such negative effects as the Hawthorne effect and we are provided with a permanent record of the experiment. In order to ensure that participants' task performance during the experiments is as representative of real-life as possible, think-aloud protocols are to be avoided as they can affect the way tasks are performed. Retrospective protocols are of limited use and are really only of benefit when the purpose of the evaluation is improvement rather than quantification; in the case of this study, we are concerned with the latter.

5.2.2.2 *Survey Methods*

In the previous sections detailing the various observational methods, we examined evaluation methods which provide us with objective data about how users work with an interface. From this data we can see exactly how well users perform tasks using the system and also where any problems are. This information is, without doubt, of enormous value but it is not enough on its own. To really understand whether an interface meets users' requirements, we need to elicit subjective information from users to illustrate their attitudes to and perceptions of the system (Dix 1998:431). Indeed, as Preece (1994:628) points out, users' opinions can affect the design of an interface while their attitudes affect the acceptance of the interface in the workplace. In short, if users do not like an interface, they will not use it unless they absolutely have to.

User survey methods - also known as query techniques or subjective assessment – make it possible to target large numbers of users to obtain their opinions directly and to highlight problems which may not have been anticipated by designers or evaluators (Dix *ibid.*). There are two main types of survey technique: *interviews* and *questionnaires.*

Interviews

The process of interviewing users regarding their experiences with an interface is a direct and structured way of gathering subjective information (Dix 1998:432). Interviews can generally take one of three forms: *structured interviews*, *flexible interviews* and *semi-structured interviews.*

Structured Interviews

In a *structured interview*, the interviewer uses a fixed and predetermined series of questions which are asked in strict order. This approach allows for close control of the type of data gathered and makes it easier to analyse. By limiting the possibilities for tangential discussions and comments, structured interviews ensure that interviewers are not distracted from the true aim of the evaluation and that the desired information is obtained. This type of interview is generally easier to conduct and it is easier to analyse the results (Hill 1995:130).

However, due to their lack of flexibility, structured interviews do not allow interviewers to follow up new lines of enquiry or discover new information. Nevertheless, the nature of this study means that the information gathered from structured interviews, while of potential interest, is not essential for the purposes of the evaluation. Here we are more concerned with user attitudes and opinions than with their observations and suggestions as to how to improve the interface.

Flexible Interviews

Flexible interviews, on the other hand, have no set questions, only a number of set topics to guide the interviewer. With no set questions, the interviewer is free to follow any line of questioning that is of interest. This method provides much more information than the structured interview but this also means that the data will be more difficult to analyse. In addition, this type of interview requires experienced interviewers to put interviewees at ease and a considerable amount of effort to analyse the data (Preece 1994: 628-629). In addition, the sheer volume of information generated may prove problematic and it is also possible that interesting new lines of questioning may result in key information being omitted or forgotten.

Semi-Structured Interviews

The *semi-structured interview* is a hybrid technique which lies between structured and flexible interviews. This type of interview features a series of questions which can be asked in any order, combined or even omitted as necessary. This gives the interviewer a significant level of freedom to explore new lines of enquiry which may emerge during the interview while still ensuring that the required information is gathered. However, this type of interview, like the flexible interview requires experienced interviewers and a considerable amount of effort to analyse the data (Preece 1994: 628-629). The availability of experienced interviewers is, therefore, a major factor when considering this approach. As Downton (1991:337) points out, interviewing for surveys is a skill in itself which requires training. This can make flexible interviews impractical in many cases. In our case, the problems of finding skilled interviewers coupled with the intensive resource requirements (e.g. time, finance, processing etc.) make interviews unsuitable for our purposes.

Questionnaires

Questionnaires are less flexible than interviews but they take less time to administer, can reach a larger audience and the resulting data is easier to analyse (Dix 1998:432; Hill 1995:130; Downton 1991:334). However, a questionnaire, because of its static nature, needs to be carefully worded to ensure that participants understand the questions fully and provide the desired information.

Questionnaires can be either self-administered or interviewer-administered. Self-administered questionnaires are completed by users alone and are frequently posted out to users. While the staff resources are generally less than for interviewer-administered questionnaires, this particular benefit is outweighed by the fact that clarification cannot be given to users. Consequently, there may be instances where users either do not answer questions or misunderstand them and give a false or misleading answer.

An additional problem with self-administered questionnaires is the frequently poor response rates (Faulkner 1998: 118). Downton (1991:335) cites a response rate of less than 40% for postal questionnaires. This can result in extreme bias of results because of small, unrepresentative subject populations (*ibid.*).

Self-administered questionnaires also place greater pressure on the designers of the questionnaire to produce absolutely clear and unambiguous questions in order to ensure that users understand all of the questions. In order to do this a continual process of design, evaluation and redesign needs to be carried out until there is absolutely no room for confusion (Faulkner 1998:117). This would undoubtedly result in a lengthy and time-consuming process which would place additional demands on time, finances and other resources.

Interviewer-administered questionnaires involve an interviewer asking the questions and completing the questionnaire with the user's responses. Although this method requires the availability of an interviewer, the interviewer does not need the same level of skill or experience as for interviews. Furthermore, interviewer-administered questionnaires make it possible to better control the data gathering process (Downton 1991:335) and any confusion as regards unclear questions can be clarified immediately. In addition, the use of interviewers ensures that the poor response rates associated with self-administered questionnaires are avoided. Despite the need for interviewers, interviewer-administered questionnaires are preferable for the purposes of this study because of the higher response rates and the fact that there is no need to continuously design, test and refine the questionnaire.

Types of Questions

There are three basic types of questions that can be used in a questionnaire: *factual*, *opinion* and *attitude*.

Factual questions, as the name suggests, ask about facts and information which is observable and public but which would be too time consuming or inconvenient to obtain any other way (Kirakowski [unknown]:3). Examples of such questions might include asking users which software packages they have experience of, how frequently people use a particular piece of software, how long people have been using a PC on average etc.

Opinion questions ask respondents to say what they think about something. Such questions aim to determine how popular something is or whether respondents like something or prefer one thing over another (*ibid.*).

Attitude questions aim to uncover a respondent's "internal response to events and situations in their lives" (*ibid.*). Such questions seek to find out what users' attitudes are to working with an interface. From such questions, Kirakowski (*ibid.*) maintains that users' attitudes to working with a product can be categorised as follows:

- users' feelings of being efficient
- the degree to which the users like the system or interface
- how helpful the users feel the system or interface is
- the extent to which users feel in control of the interaction
- the extent to which users feel that they can learn more about the system by using it.

Presentation of Questions

In addition to the broad types of questions outlined above, there are two fundamental styles of question which can be used to elicit the desired information: *open* and *closed* questions. Open questions ask users to provide answers in their own words. Closed questions ask users to select their answer from a predefined list of options.

Both styles of question have their own distinct advantages and disadvantages. For example, open questions provide a wealth of information covering a broad range of issues but they are difficult to analyse on account of the sheer volume of data produced and the variations in the style and content of responses (Dix 1998:433). Faulkner (1998:117) shares this opinion and says "the problem with open questions is that they can produce too much data which is not easily analysed".

Closed questions, on the other hand, are generally easier to analyse than open questions and they allow evaluators to focus on specific data which can be compared against other data. Furthermore, the data obtained from closed questions is more predictable and requires less interpretation (Downton 1991:336). However, such questions need to be carefully phrased in order to elicit the precise information sought.

Types of Closed Questions

There is a range of ways in which closed questions can be structured in order to elicit a particular form of data. These range from simple checklists to more complex multi-point scales and semantic differential scales.

Checklists are the simplest form of closed question and they ask for basic responses to specific questions. This type of closed question is ideal for factual information such as which software packages respondents have used etc.

Multi-point or *scalar rated questions* ask respondents to rate a specific statement between two polar opposites. This approach is suitable for determining user opinions.

Likert scales are similar to multi-point scales but in this case, respondents are asked to indicate the extent to which they agree or disagree with a statement. According to Kirakowski (unknown:8), it is necessary to prove "that each item of the questionnaire has a similar psychological 'weight' in the respondent's mind". In order to prove the reliability of such scales, some form of psychometric evaluation is necessary (*ibid.*).

Ranked order questions dispense with scales and ask respondents to number, in order of preference, a series of options. This approach is best used with a limited number of options, otherwise respondents may give arbitrary answers (Preece 1994:633).

Semantic differential questions are similar to Likert scales but they ask respondents to rate an interface on a scale between two diametrically opposed adjectives, e.g. clear-confusing or interesting-boring (Preece 1994:632).

Multiple-choice questions offer a range of explicit responses and respondents are asked to select either one or more of these options.

A concern with questions that involve some form of scale is the granularity or number of rating points provided to the respondent. This relates to the level of detail an evaluator wants to achieve in the collated data. If a broad general idea of preferences or opinions is required, a simple three-point scale is adequate. However, if a more detailed breakdown of preferences and opinions is required, a greater number of rating points can be used. However, simply adding more and more points on the scale can prove counter-productive. If we use a ten-point scale, it is likely that some respondents may find it difficult to differentiate between any two adjacent points on the scale with the result that they may arbitrarily pick points (Dix 1998:433).

On a related note, Kirakowski (unknown:10) raises the question of whether to use an odd or even number of rating points on the scale. The reasoning behind this is that with odd-numbered scales where the central point corresponds to neutral opinions or undecided users, respondents may "go on auto-pilot" and select neutral points without giving any real thought to their choice. As a way of preventing this, an even numbers of option can be used to "force" respondents "to go one way or another" (*ibid.*). This is, according to Kirakowski, unhelpful in that it does not cater for respondents who genuinely have no preference or strong opinions. It is also common for respondents to randomly pick alternate options from the two middle options.

Developing a Questionnaire

Kirakowski (unknown:6) maintains that developing a questionnaire requires "a lot of time, patience and resources". In addition, considerable knowledge of psychological measurement and statistics is essential. Kirakowski also maintains that unless all of these prerequisites are met, serious questions arise as to the validity and reliability of the questionnaire. As a result, he recommends that a pre-designed, pre-tested questionnaire should be used instead. In view of this, we will examine a number of commonly available usability questionnaires and select a suitable one. These range from freely available, public-domain versions to sophisticated commercial varieties. If necessary, we will modify the questionnaire to suit our purposes.

A number of the available models are simply unsuitable for the purposes of this study, e.g. *WAMMI*[3] which is designed for testing websites. Others, such as *IsoNorm*[4] (developed by Jochim Pümper on the basis of ISO9241-10) are only available in languages other than English which could not be understood by participants. Of the remaining models, a number of criteria (see Table 14) were used to select the most appropriate one.

■ cost	■ reliability
■ data validation & standardisation	■ flexibility/customisation

Table 14: Criteria for Examining Questionnaires

[3] see http://www.wammi.com
[4] see http://www.sozialnetz-hessen.de/ergo-online/Software/Isonorm-Workshop.htm

The first such model reviewed was the *Computer System Usability Questionnaire* or CSUQ[5]. This questionnaire is available free of charge and incorporates psychometric reliability properties. However, the test is run using a web page which is submitted to another server for processing. While this removes the effort of processing data, it also leaves the test administrator without a permanent record of the answers provided. Another problem with this is the fact that questions are phrased in a way that could lead or prompt specific responses. Obviously, this can distort responses and lead to unreliable data. Furthermore, the questionnaire does not refer to documentation and it cannot be customised or modified.

Another questionnaire was the *Software Usability Measurement Inventory* or SUMI[6] developed by the *Human Factors Research Group* at University College Cork. This questionnaire overcomes the problems of CSUQ in that questions are worded in a variety of ways so as to avoid leading respondents. SUMI is supplemented with a data analysis and reporting service which eliminates the need for test administrators to perform complex calculations. However, samples of SUMI available on the Internet fail to include documentation. Above all, the high cost of SUMI means that it was not a financially viable option for this study.

The questionnaire selected for this study was the *Questionnaire for User Interaction Satisfaction* or QUIS[7] developed by Kent Norman and his research team at the *University of Maryland at College Park*. Using a 9-point Likert scale, QUIS features a section to determine overall levels of satisfaction and hierarchical sections which examine satisfaction with specific aspects of the interface (see Table 15).

■ screen factors	■ learning factors
■ technical manuals	■ multimedia
■ virtual environments	■ software installation
■ terminology and system feedback	■ system capabilities
■ online tutorials	■ voice recognition
■ Internet access	

Table 15: QUIS Question Categories

[5] see http://www.acm.org/~perlman/question.cgi?form=CSUQ
[6] see http://sumi.ucc.ie
[7] see http://www.cs.umd.edu/hcil/quis/

What makes this model so attractive is that is can be modified and customised to suit the needs of a particular study. Thus entire blocks of questions can be omitted. Just as importantly, all questions are phrased in a neutral way which neither encourages nor discourages a particular response. On a practical level, QUIS can be administered as a web-based questionnaire (although this requires a specially configured web server) or as a paper-based questionnaire.

QUIS also comes with instructions for analysing and processing data. Indeed, this is reinforced by references to other research projects which have used QUIS. From a cost point of view, QUIS is comparable to other questionnaires such as SUMI but there is an option of obtaining a student site licence which costs USD$50 (at time of writing).

5.2.3 Participants in a Usability Evaluation

As we have already discussed above, the aim of usability evaluation is to see how easy users find a system to use. It is obvious, therefore, that the participants in a usability evaluation should reflect the real users as accurately as possible. Consequently, finding and recruiting participants can be a complex process which must be planned carefully (Downton 1991:340). There are two key factors to be remembered when selecting participants for a usability evaluation: *who they are* and *how many of them are needed.*

5.2.3.1 *Background of Participants*

In asking who the participants are, we are concerned with those characteristics of the participants which may have some bearing on the evaluation. We need to ask questions such as what do they know? What skills and experience do they have? What is their age, gender, background and level of education? But for there to be any point in gathering this information, we must first know something about the real users of the system. In order to understand the characteristics of real users, we need user profiles so that we can select suitable participants to represent them in the usability evaluation (Dumas & Redish 1993:120).

Dumas & Redish (1993:122) propose a method of creating user profiles which involves usability experts working in conjunction with subject specialists or the actual users to define precise information relating to the users' backgrounds. Such information includes:

- work experience, e.g. job description, length of service, experience with particular tasks
- general computer experience, e.g. length of experience with specific types of applications, frequency of use etc.
- specific experience with operating systems, hardware (e.g. mouse or keyboard) etc.
- experience with this and similar products

Although Dumas & Redish do not explicitly say so, it is, of course, conceivable that this information could be gathered or supplemented by means of interviews or questionnaires. Alternatively, this information could be elicited, for example, from a manager or someone who supervises and recruits real users or possibly even someone who knows the technical skills of users, e.g. technical support engineers.

Dumas & Redish present a sample user profile which consists of five basic headings. Although this profile is intended for screening participants, it is, of course, also suitable for creating profiles of the real users.

- product name
- general characteristics of user population
- characteristics of users that are relevant to the test
- which user characteristics should all users have and how will you define them?
- which user characteristics will vary in the test and how will you define them?

We can expand this basic profile to produce a user profile questionnaire as shown in *Appendix A*.

Once this profile has been created for users, it is necessary to create a similar profile for each of the potential participants. With this information, selecting those participants who most closely match the real users is a relatively straight-forward process. The data obtained for participants can then be used to distribute participants with varying skills across a number of subgroups (Faulkner 1998:115). In our case, there will be two sub-groups: a control group and an experimental group. This is important because if one group has more participants with a particular type of skill relevant to the test than another group, the results may be distorted (*ibid.*).

5.2.3.2 Numbers of Participants

A critical element determining the success of any experiment in terms of validity and accuracy is the numbers of participants involved. Quite simply, there have to be enough participants in the test to ensure that the results are truly representative of the real users as a whole and that the results are not just the idiosyncratic behaviour of participants (Dumas & Redish 1993:128). Nevertheless, it would be impractical and indeed extremely difficult to use vast numbers of participants. Obviously some form of compromise is needed.

Nielsen (2001) maintains that for the purposes of usability testing in industry, 3-5 participants are generally sufficient in order to gain an insight into the usability of a product. Nielsen makes the point that after the fifth participant, most of the problems have already been discovered and any subsequent participants will only repeat and confirm what is already known. This is echoed by Dumas & Redish (1993:128) who say that "after you've seen several people make the same mistake, you don't need to see it a 10^{th} or 20^{th} or 50^{th} time". As we are comparing two versions of an interface (i.e. one user guide with iconic linkage and one without) we will, of course, need twice the number of participants. This fits in with claims by Dumas & Redish that usability tests generally involve 6-12 participants divided among a number of subgroups.

5.3 Usability Evaluation Procedures

In the previous sections we discussed the various approaches, methods and techniques used in usability evaluation. This section examines how these various factors are implemented as part of a usability evaluation. The purpose of this section is to discuss how and why these tools are used in an evaluation. The procedures for conducting evaluations will also be discussed with reference to a number of case studies which relate directly to this study.

5.3.1 Practical Usability Evaluation

While having a firm understanding of the methods and tools described in previous sections is, without doubt, essential in conducting usability evaluations, the success or failure of an evaluation depends on a variety of preparations being carried out. Without these preparations, the test will stumble awkwardly on, with no clear goal, purpose, transparency or logic. In the following sections, we discuss the various practical tasks that form a central part of usability evaluation.

5.3.1.1 Measuring Usability

As we discussed in Chapter 3, usability involves not only how well users can use an interface, but also how users perceive the interface and what their subjective opinions of the interface are (Dumas & Redish 1993:184).

We also know that the usability of an interface can be measured in absolute terms on the basis of predefined performance criteria or it can be compared against that of another interface (see page 157). However, the term performance is vast and can refer to a multitude of elements and factors of interface usage and the execution of tasks. Unless we define which aspects of performance we want to measure, it will be difficult, if not impossible, to determine the usability of an interface.

Wixon & Wilson (1997:664) present a list of what they term "usability attributes" which are characteristics of an interface. These attributes can be used to categorise and quantify the various facets of an interface's performance.

The attributes proposed by Wixon & Wilson include:

- usefulness
- efficiency
- memorability
- advanced feature usage
- flexibility
- learnability
- error rates
- first impressions
- satisfaction or likeability
- evolvability

It is apparent that there are some questionable inclusions in this list such as "usefulness" which, as we have already discovered (see page 53), is quite separate from usability in that it is a social or commercial factor which does not effect how well users can use something.

On the basis of these categories it is possible to compile a list of measurement criteria for use in usability evaluations. These criteria make it possible to count or measure individual behaviours on the part of the user. Dumas & Redish (1993:184) point out that counting instances of user behaviour requires careful attention but, importantly, that it does not require judgmental decisions; either the event occurred or it did not provided, of course, such events are observable and discrete.

Wixon & Wilson (1997:666) provide the following list of measurement criteria:

- time to complete a task
- number of subtasks completed
- time needed to complete a task after a specified period of time away from the system
- number of steps required to complete a task
- number of times users access documentation or technical support
- number of tasks completed
- number of errors per unit of time
- time spent recovering from errors versus time spent working productively
- number of negative reactions to interface
- number of commands or icons remembered after task completion

Dumas & Redish (1993:185) provide their own list of criteria which, although similar in certain respects to the criteria of Wixon & Wilson, are more detailed in their formulation. These criteria include:

- time to complete a task
- time spent navigating menus
- time spent in online help
- time spent finding information in manual
- time spent reading manual
- time spent recovering from errors
- the number of incorrect menu choices
- the number of incorrect choices in dialog boxes
- the number of incorrect icon choices
- the number of incorrect function key choices
- the number of other errors
- the number of repeated errors
- the number of calls to technical support or for assistance
- the number of help screens looked at
- the number of repeated visits to the same help screens
- the number of times a manual is consulted
- the number of times a quick reference card is consulted
- the number of pages looked at on each visit to the manual
- the number of searches in the index on each visit to the manual
- the number of searches in the table of contents on each visit to the manual
- observations of frustration
- observations of confusion
- observations of satisfaction

It is clear that between these lists, not to mention other similar lists such as that compiled by Preece (1994:405), there is a wide range of criteria which can be measured. Obviously, recording data for all of these criteria would be extremely time-consuming, if not overwhelming. Indeed, Wixon & Wilson (1997:667) maintain that the number of criteria should not overwhelm the test team. They go on to say that 2-3 criteria are sufficient to measure usability. In contrast, Dumas & Redish (1993:185), while acknowledging the impracticality of using all of the criteria, do not restrict the number like Wixon & Wilson. Instead, they say that not all of the criteria are applicable to each test and that only those that relate directly to the product should be used.

If we refer back to Section 4.7.6.1 where we detailed the ways in which Iconic Linkage can improve usability, we can see that the other main attributes of interest are: learnability, retention of information over time, comprehensibility, accessibility of information and speed of processing.

5.3.1.2 Usability Evaluation Case Studies

Despite being a developmental evaluation model and despite grouping print documentation together, Simpson (1990) discusses two examples of usability testing which give some useful practical tips for conducting evaluations. One of the studies involved testing online help while the other involved a computer-based tutorial (CBT).

A crucial question investigators must ask themselves, Simpson asserts (1990:42), is what specific data is sought. Simpson maintains that the deciding factor in choosing an evaluation method is the type of usability information needed. He proposes the following stages for any form of testing (1990:45):

- define the test question
- decide what data is needed in order to answer these questions
- select methods for getting this data
- plan how the methods should be implemented

By his own admission, this process is rarely as straightforward as it seems. Beyond this overview, however, Simpson provides little useful practical advice.

Another study, carried out by Harrison and Mancey (1998), compares two versions of an online, web-based manual and examines the optimum elapsed time before gathering users' reactions to the different designs. Rather than examining textual or content-related factors, this study compared different navigation models and their effect on usability. Although this study also treats online and print documentation identically and its objectives are dissimilar to our objectives here, it provides a useful insight into procedures for gathering data using user surveys.

As a way of testing how well users learn and remember information from a manual, the study used a series of questions based on the information contained in the manual. There were eight groups of twelve questions which took the form of cloze tests which could be answered with a single one, two or three word response. Such a method could be used to test the notion that usable texts promote the retention of information over time (see Section 4.7.6.1).

Interestingly, this study also utilised a written script for researchers to follow during tests to ensure consistency for all subjects. The authors do not, however, give any details of the actual tasks involved or the efficiency and error criteria employed (if any). This can be attributed to the fact that the aim of the study was not actually concerned with measuring usability *per se*.

The main finding of the study was that the length of time a user spends working with a product before being asked to give an evaluation affects the final evaluation. However, the authors found that evaluations stabilised after working with the product for 15 minutes. This also shows that think-aloud protocols, argued to be more accurate because of the immediacy of responses, are unnecessary for the purposes of gauging user satisfaction and opinions as there is no pressing need for immediate feedback.

Teague *et al.* (2001) conducted a series of tests at Intel Corp. in Oregon with the similar aim of establishing whether there were significant differences when users are asked to rate ease of use and satisfaction during and after tests. A total of 28 subjects were recruited to perform a variety of tasks using a range of commercial websites. Tested individually, subjects in the two groups were asked questions at either 30 or 120 second intervals while performing the tasks. The questions were based on seven-point Likert scales and subjects had to answer each question orally during the task. After the task, the subjects were asked to answer the questions again in writing. A third group, who did not answer questions during the task, only answered the questions in writing.

The results of this study appeared to indicate that post-task responses were "inflated" and that users gave more honest and representative answers during the task. Not only is this finding in conflict with Harrison & Mancey (1998), but it can be argued that there were other psychological factors at work which can account for this phenomenon. According to various social psychologists, most notably Asch (1956) and Sherif (1937), conformity and the desire to conform and be accepted can frequently cause people to give "false" or less than truthful answers, even though they do not reflect what a person actually thinks. This desire to conform is most pronounced when subjects are asked to publicly verbalise their responses. In comparison, the need to conform is less obvious where subjects are asked to write down their responses in private (Deutsch & Gerard 1955). Thus, it is reasonable to assume that the "inflated" results in the post-task survey are actually more indicative of the subjects' real ratings than the verbal, concurrent ratings. In any case, it can also be argued that subjects' responses only stabilised after completing the tasks (as mentioned previously by Harrison & Mancey 1998). It is possible that, for whatever reason, the subjects were (unwittingly) biased into giving negative answers because they thought that that was what was expected of them.

Another possible explanation can be deduced from the finding that subjects who only answered questions in the post-task evaluation performed their tasks more quickly than the concurrent groups. The concurrent groups took on average 15% longer to perform the tasks and found the tasks significantly less enjoyable. We can attribute this to the regular interruption and distraction caused by the questions and the subsequent need to refocus on the task at hand. Such activities require additional cognitive effort and as such increase the workload, fatigue and stress for subjects. It is clear, therefore, that post-task evaluation appears to be a more considerate and indeed accurate means of data collection than any concurrent form of questioning.

In a study conducted to assess whether the results of a usability test using test subjects can be used to reliably predict actual satisfaction among real users, Notess and Swan (2003) embark upon an approach which is fundamentally flawed from the outset. The study involved three rounds of tests: a baseline session and two comparative sessions.

The baseline session involved 30 subjects testing an existing version of a digital music library application. These subjects were students recruited at random as they entered a computer laboratory where the software was installed. The students were originally planning on using the software anyway and therefore represented real users of an existing version of the software. After they had completed their tasks (it is unclear whether the tasks were prescribed by the researchers or whether the students simply carried out their own work), the subjects completed a QUIS-style questionnaire.

The following session, called the subject satisfaction test, involved ten people and was designed to evaluate satisfaction with a new version of the software which was under development. The third session was called the user satisfaction test and was intended to assess the attitudes of some 12 (out of a total of 30 participants) real users. Rather bizarrely, the software used in the user satisfaction test was modified midway during the test. Such an event instantly renders the results useless from an experimental point of view. This however, was not the only flaw in the experimental design. In addition, the three groups of participants contained different numbers of people and none of the participants were profiled to ensure representativeness of the real intended users. The tasks carried out by the participants were not prescribed nor were they standardised in any of the groups. According to the authors, one of the tasks in one group in the user test was significantly more difficult than the tasks in the other tests.

Furthermore, no efficiency or performance criteria were established. Unsurprisingly, the data obtained from the questionnaire failed to produce any statistically significant results and no conclusions could be drawn from the data. This study provides a valuable insight into the type of problems that can damage the effectiveness and validity of an experiment.

In contrast to the preceding studies, Zirinsky (1987) provides a detailed and useful discussion of usability evaluation aimed specifically at printed documentation. It is not surprising, therefore, given the discussion in Section 1.1, that the study dates from the late 1980s. Zirinsky starts off by stating that in a usability test involving users, we want users to tell us what they dislike about a product, not what they like (1987:62). The role of testers, he continues is to find problems, not to impress researchers with expert performance. A similar point is made by Redish and Dumas (1993:276) who emphasise that users should realise that they are not being tested.

Zirinsky provides a number of recommendations for those preparing to conduct a usability test. The first of these is that all of the test materials (1987:62) should be edited. As part of the editing process, it is essential that there are no typographical errors, style inconsistencies, grammatical or punctuation errors which can distract subjects or even cause them to doubt the validity of the technical material presented. This leads on to checking the document for both technical content and linguistic accuracy. Zirinsky maintains that a manual will improve by no more than 20% as a result of a review, so the better the quality of the product to start with, the better it will be after being reviewed and tested.

As regards actually conducting the test, Zirinsky asserts that users should remain objective and should be fully briefed about the product and their role in the test. They should only be provided with enough information to ensure that they fully understand what is expected of them. Subjects should not be told what the researchers are looking for, i.e. they should be told that they are looking to see which of two versions of a user guide is better, not that we are looking to see what effect repetition has on a document's usability. Furthermore, subjects must be made to feel relaxed and confident enough to make constructive criticisms and comments regarding the document.

It is clear from the previous studies that many of the approaches simply do not apply completely to this study, even though several of the studies provide useful practical pointers. Of the literature reviewed, only two studies specifically set out to conduct comparative usability tests on print documentation where the object is to gauge the effect of a single, non-technical variable. As such, these studies provide a broad framework or model for conducting an empirical study to test the effect of Iconic Linkage. The first of these, conducted by Foss *et al.* in 1981, aimed to improve usability and accelerate learning by examining the use of supplementary information and the effect of restructuring a user guide. The second study was conducted by Sullivan and Chapanis in 1983 and was concerned specifically with re-engineering a user guide to take into account best practice in terms of technical communication and human factors. The following sections describe these studies in detail.

Foss et al. 1981

Basing their work on the claim that previous work in the area of text accessibility and usability was vague and often contradictory, the authors set out to "understand better the acquisition, representation, and utilization of knowledge by novice or occasional users" of software and to "test some ideas derived from current views of memory and attention" (Foss *et al.* 1981:332).

Hypothesis

The basic hypothesis is that users learn more effectively when they understand what they are doing. To test this, a comparative experiment was carried out using two versions of a computer manual: both versions were essentially identical in content but one was restructured to present information in order of progressing complexity. Two groups were formed and each was given one of the two versions of the manual.

In addition to the original and revised manuals, the authors produced what they call an "*Advance Organiser*". This document consisted of an explanation and description of the basic characteristics of the software being used. It did this in general terms describing key parts of the software without referring to specific elements of the software. The *Advance Organiser* was given to half of the subjects in each of the two groups.

PROCEDURE

In conducting the experiment, which took three hours per subject, subjects were first given general information about the experiment and their typing speeds were measured. Selected subjects were given the *Advance Organiser* and told to study it. Subjects were then given one of the two manuals and told to study it for 15-30 minutes. Once this had been completed, a booklet containing nine text editing exercises was distributed to subjects.

The subjects were told to rely as much as possible on the manuals and that they could only ask the experimenter as a last resort. During the course of the experiment, the experimenter remained in the room noting certain aspects of the subjects' behaviours such as the amount of time they spent performing each task, the number of interactions between the subject and the experimenter as well as the number of tasks completed within the allotted time. Although subjects were told not to ask the experimenter for help, some did in fact ask for help. Such instances were dealt with according to a strict procedure. Firstly, subjects were told that the information they required was contained in the manual. If this was unsuccessful, the experimenter pointed out the specific chapter in the manual. If the subject still experienced difficulties, the experimenter gave explicit instructions.

Such an approach may seem inappropriate when the purpose of the experiment is to determine the performance of subjects using the manual. However, if we look at this in a wider context we can see that if a manual is effective, there should be no need for subjects to ask questions. Thus, a subject being forced to ask certain types of question indicates some form of problem in the manual. Sometimes during the experiment, the experimenter intervened when it was apparent that the subject was struggling to complete tasks, e.g. the subject sat for long periods without accomplishing anything.

In addition to the manual recording of subjects' behaviours by the experimenter, the experiment also utilised a rudimentary (by today's standards at least) keystroke logging application which recorded the type and number of commands used to perform each task.

On the basis of the data collection techniques employed in the experiment, it was possible to analyse performance on the basis of the following criteria:

- number of tasks completed
- average time to complete a task
- average number of commands used to complete a task
- average number of verbal interactions during a task
- number of errors at the point where the subject said the task was complete

The results of the experiment showed that the organisation of a manual can dramatically improve user performance. It also showed that providing advance information allowed users to "scaffold"[8] their learning, thereby making learning more efficient. While this study does not relate directly to the current research in that it involves restructuring the sequence of information in a manual and the use of external materials, it does provide a useful insight into usability evaluation procedures involving documentation.

Another case study which relates more closely to this research in terms of goals and materials is the one carried out by Sullivan and Chapanis in 1983.

Sullivan & Chapanis 1983

Like Foss *et al.* (1981), Sullivan and Chapanis set out to investigate ways of improving the usability of computer documentation by evaluating different versions of a user guide for a text editing software application. However, this particular study differs from that of *Foss et al.* on a number of points: firstly, this study involves the creation

[8] For a more detailed discussion of scaffolding in the context of social constructivist theories of learning, see Kiraly (2000).

of an entirely new manual from scratch; secondly, the study is concerned with the content and formulation of information in the manual as opposed to the organisation of information or the use of supplementary sources of information.

According to the authors, the purpose of this study is...

> ...to improve an existing computer manual through the application of rules of document preparation, to measure the extent of the improvement and, based on that experience, to propose a general methodology for the preparation of documentation (Sullivan and Chapanis 1983:114).

The study consists of two broad components. The first component involved producing an "improved" manual on the basis of a review of literature on writing instructions. The second stage involved conducting a comparative analysis of the original manual and the new version of the manual. Both of these stages involved an element of experimental evaluation and are described in greater detail below.

Preparation of New Manual

Beginning with a review of literature on writing instructions, the authors compiled a list of guidelines for making instructions more readable (Sullivan and Chapanis 1983:115). These guidelines included:

- use simple, familiar language
- use short, active positive sentences
- sequence events in the text in the order they are to be performed
- provide complete and specific descriptions of actions
- describe one thing at a time
- use headings and subheadings to identify sections of the instructions
- use lists rather than large blocks of prose

Next, a task analysis of the system was carried out and potential difficulties were noted. This task analysis involved listing operator tasks and responsibilities, determining normal task sequences and the sequences of operations as well as examining cautions and warnings. The task analysis also described the decision-making operations and determined abnormal task sequences.

On the basis of this task analysis, the authors set about preparing a first draft of the new manual. The manual was divided into two sections. The first section consisted of an introduction to the software and a tutorial which guided users through the basic software operations. The second section of the manual consisted of a reference section divided into functional groups along the same lines as the tutorial.

A walk-through test was then conducted to identify areas that needed improvement and areas where information had been omitted. The necessary changes were incorporated into the second draft of the manual.

Next, a user edit was conducted using five subjects. These subjects were secretaries aged between 20-35 years of age, all of whom had previous typing and editing experience and who had a typing speed of at least 30 words per minute. The subjects were recruited through a temporary staff agency. Each subject performed a series of tasks using the software and the manual and the difficulties they experienced were noted using a variety of data collection methods. The first of these methods involved linking an additional, slave monitor to the subject's computer monitor. A video camera was used to record the events shown on the slave monitor. As each subject performed the tasks, the experimenter, who remained in the room with the subject, noted any difficulties on paper and answered any questions the subject had.

After each subject had completed the tasks, the manual was modified to rectify the problems which emerged during the test. This continued until users could perform all of the tasks with only one or two questions. This was achieved after four subjects had participated.

When subjects has completed the tasks, they were asked to complete a 10 question, 7-point Likert scale attitude questionnaire. This was done to pre-test the questionnaire for the second part of the study.

The video data was also analysed to identify problems and difficulties encountered. A final edit of the manual was then carried out. This final version of the manual featuring all of the necessary modifications was then used in the second or main part of the study.

Comparative Analysis of Old and New Manuals

The main study involved comparing user performance using the old and new manuals in a controlled laboratory experiment. Twelve subjects of a similar background to those used in the first part of the study were recruited by means of a newspaper advertisement. The age range of these subjects was between 20-50 years of age.

In contrast with the previous test, the authors dispensed with the video camera and instead opted for a modified version of the text editing software which created time-stamped logs of subjects' input. The events in the test room, e.g. questions from subjects and interventions by the experimenter were recorded on audio tape. Instead of three tasks like the first test, the main study involved just two of the original tasks because of time constraints.

In preparing the two versions of the manual, the authors edited the old manual to bring certain terminology in line with the more modern hardware being used in the experiment. The old manual was also reprinted on the same printer as the new one to eliminate any differences in the appearance of fonts and layout. Furthermore, sections of the old manual which were not contained in the new manual, such as lesser used formatting codes and control functions, were deleted. This is similar to the Foss et *al.* study where the authors ensured that the overall content and layout of both manuals was essentially the same.

PROCEDURE

The experiment was carried out over two sessions for each subject. In the first session, subjects familiarised themselves with the computer using a keyboard learning program. In the second session, subjects were randomly chosen to use either the old manual or the new manual. The first task was allocated 2.5 hours but subjects were informed that they should not feel pressurised because most people would be unable to complete all of the tasks within the allotted time. When this task was completed, subjects were given a 15 minute break before commencing the second task which lasted for an hour.

During the tasks, the subjects were told that if they encountered a problem, they should first try to solve it themselves. They could only ask the experimenter if they really could not solve a problem themselves. If subjects asked general questions, the experiment referred to the manual. If subjects asked specific questions, the experimenter gave specific answers.

When the tests were completed and all data had been gathered, performance was measured on the basis of the following criteria (Sullivan & Chapanis 1983:119):

- quality of text produced
- number of different commands used successfully
- number of questions asked
- type of questions asked

Recognising the importance of subjective user attitudes to overall usability levels, the authors used a post-test questionnaire to determine subjects' attitudes to the manual, the software and the tasks. However, attitudes to the software and tasks were not related to which manual was used (*ibid.*:122). In contrast to the initial questionnaire, the one used here consisted of thirteen 7-point Likert scale questions. Four of these questions related to the manual while the remainder related to the software and the tasks.

FINDINGS

The authors found that the subjects who used the new manual performed significantly better in the tasks than the subjects who had used the old manual and that the new manual group were able to use more commands successfully without the assistance of the experimenter. They also found that the group using the old manual asked more than four times as many questions as the other group and that this indicated serious problems with the old manual.

The questions asked by subjects during the tasks were categorised as follows:

- manual organisation problems
- global misunderstanding
- not knowing what command to use
- failing to distinguish between writing and editing modes
- not knowing how to use a command or code
- system-dependent problems
- task clarification questions
- negative comments

The results of the test showed that in all categories except category 4, the subjects who used the new manual asked far fewer questions than the other group. In category 4, the number of questions remained constant.

The overall findings of the study showed that it is possible to improve the usability of user documentation through a combination of "local" production rules and iterative production processes. The authors also maintain that the involvement of potential users in the development process is essential.

5.4 Experiment to Test The Impact of Iconic Linkage

Irrespective of the amount of theoretical evidence contained in the literature on HCI, technical communication and cognitive psychology to suggest that Iconic Linkage will improve the usability of a software user guide, the only reliable way to establish whether this is true is to put it into practice. For this reason it is essential to conduct some form of empirical experiment to determine what, if any, effect the introduction of Iconic Linkage into a text will have on the text's usability. More specifically, an empirical study is needed to establish whether the presence of Iconic Linkage in a text improves the usability of the text and thus, its quality, or whether it has no effect whatsoever.

Given the fact that usability relates to the ease with which people can use something, the logical starting point is to develop a user-based usability study. Since we are interested in obtaining a detailed insight into usability consisting of both how well users work with a user guide and what their opinions are, we will opt for a combination of subjective and objective quantitative testing. The following sections describe the preparations, rationale and methods used in conducting the pilot and main study.

5.4.1 Pilot Study

The purpose of this study was to test the methods, materials and procedures for determining the effect of Iconic Linkage on the usability of a software user guide. Although Iconic Linkage can be implemented in both monolingual text production situations and in translation situations, the study was restricted to examining Iconic Linkage in a single language environment to eliminate the potentially confounding influence of an additional language and the translation process. In addition to reflecting the production of user guides in a single language, this also reflects the introduction of Iconic Linkage during the editing, proofing and correction stages of both original language documents and translations.

For the purposes of this study, usability (or more specifically, factors which indicate usability problems) was measured on the basis of the following quantifiable criteria (drawing on those discussed on page 175):

■ Subtasks completed	■ Tasks Completed
■ Times user guide used	■ Negative reactions to system
■ Incorrect icon choices	■ Commands and icons remembered afterwards
■ Verbal interactions/questions during tasks	■ Incorrect menu choices
■ Searches in TOC each time user guide used	■ Searches in the index each time user guide used
■ Observations of confusion	■ Observations of frustration
■ Errors at point where subject thinks task have been completed	■ Observations of satisfaction

Table 16: Pilot Study Usability Criteria

It was also decided to record the times for each of the following criteria:

■ Completion of tasks	■ Recovering from errors
■ Navigating menus	■ Using online help

In addition to these objectively quantifiable criteria, a subjective user satisfaction questionnaire and a post-test survey were also administered to examine usability in terms of user satisfaction and retention of information over time.

The pilot study was carried out in four major stages:

1. Preparations
2. Familiarisation
3. Testing
4. Post Test Surveys

5.4.1.1 Preparations

Preparations for the study involved the creation of a wide range of documents, forms and other materials including the two versions of the user guide. This stage also involved profiling typical users and recruiting participants.

Preparing the User Guides

The single most time-consuming and labour-intensive part of the preparations involved sourcing and editing the user guide. Firstly, a suitable user guide had to be found. The main difficulty here was finding software that was both suitable for use in the study <u>and</u> which had documentation with the appropriate content and textual features (i.e. low levels of latent Iconic Linkage). The expression "appropriate content and textual features" encompasses a number of different factors. Obviously, the user guide had to be of an acceptable professional standard from a quality perspective. Quality was assessed on the basis of completeness of information, spelling, readability, layout etc.

Secondly, and perhaps most importantly, Iconic Linkage is only of use where the same or similar information is presented at various points in the text. The challenge here was to find a user guide where the instances of non-isomorphic, but semantically identical text were not so closely packed together as to render the learning aspect of Iconic Linkage irrelevant (e.g. identical information in consecutive sentences) yet not so far apart as to necessitate reading hundreds of pages of text.

The nature of the software itself plays a crucial role in the presence of these features in that it is simply not possible to create full Iconic Linkage in the text where there is no similarity in the functions or procedures in the software.

Selected Test Software

The user guide selected for use in the study was an extract from the documentation for the *DigiTake* parliamentary recording suite developed by *DigiTake Software Systems Ltd.* I had previously been contracted by this company to translate various documents for this product. As a result, I was very familiar with both the software and the documentation. The company was more than happy for me to use its software and documentation and also provided invaluable technical assistance in using the product as part of this study. DigiTake is a suite of digital audio recording applications used to record the proceedings of meetings, debates etc. in digital audio format. This software is designed for use in venues such as parliamentary chambers, large meetings, tribunals etc. and is currently in use in the Irish Parliament (Dáil Éireann).

The extract used deals with an application called DigiLog. This application is used in conjunction with the *DigiTake* digital recording package and is used by parliamentary reporters to log the proceedings of meetings etc. As each speaker rises and begins speaking, the reporter types the speaker's name either manually or using a shortcut and then types the first 4-5 words spoken. Once this is done, the reporter waits until the next speaker takes the floor and then repeats the process. This is repeated for each speaker. In effect, the reporter logs the turn-taking in a meeting to produce a log which serves as an outline when preparing the full transcript of the proceedings at a later stage. The study emulated this procedure with the participants playing the role of the parliamentary reporter.

This software was regarded as particularly appropriate because, although it is based almost exclusively on industry-standard technology and uses word-processing technology familiar to most users, it is designed for a very specific area of application. Consequently the procedures, rather than the technology, are quite specific and specialised and are unlikely to be familiar to participants. This is important in order to ensure that participants are relying on the user guide and not on their own prior knowledge or problem-solving skills. The familiarity of the word-processing environment serves to provide reassurance for participants and also to eliminate the need for more detailed profiling of participants' previous knowledge. Another advantage of *DigiLog* is that various steps and procedures are common to different processes. This means that the likelihood of similar information occurring in the user guide is high and as such, the user guide is more likely to benefit from iconic linkage.

Versions of User Guides

Once the software and user guide were selected, the first step was to produce two versions of the user guide: one with Iconic Linkage and the other without Iconic Linkage.

The first stage of this process was to create a copy of the original user guide and ensure that both versions were identical in terms of fonts, layout and graphics.

The format of the documents was updated to make the information even clearer. Given the fact that only an extract of the complete user guide was going to be used and that the available time was limited, additional graphics were included to provide greater clarification for certain sections of the text where otherwise users would refer to the system or another section of the user guide.

The next stage involved rewriting one version of the user guide to introduce Iconic Linkage. In practice, there are a number of ways of introducing Iconic Linkage, including the use of *style guides* and *text processing software.*

Style Guides

As described in Chapter 4, style guides are a good way of implementing HCI principles. Style guides also represent an excellent way of restricting the possible ways in which information can be formulated much in the same way as controlled language does (Power *et al.* 2003:115). In effect, this facilitates the introduction of Iconic Linkage by creating an environment where isomorphic formulations are preferred. So, to implement Iconic Linkage we would first need to develop a reasonably comprehensive style guide which specifies the ways in which information can be formulated. There are, however, certain limitations in the use of style guides. It may not always be possible to develop a style guide because this is itself a lengthy and time-consuming process. As Dumas & Redish maintain, any of the publicly or commercially available style guides can be used to eliminate this problem (1993:60).

Another potential problem with style guides is that if Iconic Linkage is to be implemented on a large scale in a text, a large number of rules will be needed. This introduces the problem of learning, memorising, practising and consistently using the rules. Depending on the number of rules, it may be impossible for one person to remember them all and then to use them consistently without some form of tool to ensure adherence to the style rules (see Schwitter *et al.* 2003). Ultimately then, the best course of action is, perhaps, to opt for a fairly limited style guide which is easy to remember and implement.

Text Processing Software

Text processing software as used here refers to computer-aided translation (CAT) tools such as *Trados Translator's Workbench*, *STAR Transit* or *Atril Déjà vu.* However, it is conceivable that some other form of text storage and recognition software or text database could be used (Buchanan 1992). For the purposes of this study, Trados Translator's Workbench was used.

Under normal circumstances, CAT tools are used to help translators translate from one language to another, but they can also be used to translate intralingually or from one variety of a language to another, e.g. US to UK English. When rewriting a text using Trados, we have the source text which we will call ST and we have the edited text which we will call the *target text* or TT. As each sentence is edited and rewritten, the TT is stored along with the ST in a database. Each subsequent sentence in the text being edited is compared against the ST/TT units in the database to see if there is a match. As more and more of the text is edited, more ST/TT units are added to the database and the likelihood increases that a new sentence will at least partly match one or more of the ST/TT units already in the database.

Relating this to Iconic Linkage, while Iconic Linkage by definition refers to making isomorphic, sentences which are non-isomorphic and which cannot be detected by CAT tools because they analyse surface structure and not meaning, CAT tools can detect instances of partial matches and isomorphic sentences. That CAT tools can detect partial Iconic Linkage is clear but what makes this useful is that very often, these partial matches can sometimes be turned into full Iconic Linkage by rewriting them.

Also, CAT tools can indirectly detect non-isomorphic but semantically identical sentences thanks to the incidence of placeables. Placeables are words or phrases such as product names, toolbar icons, menu options and dialog boxes etc. that do not change in either editing or translation. As such, they will stay the same regardless of the way in which a sentence is phrased. Thus, if an existing unit contains the placeables X, Y and Z, a new sentence that has these terms may, depending on the ratio of these terms to the total number of words in the sentence, be offered as a partial match solely on the basis of these placeables. From preliminary tests using this method, fuzzy (partial) matches above approximately 60% can frequently represent instances of non-isomorphic semantic matches (see Figure 12). They can be rewritten to introduce full or partial Iconic Linkage where there was none before. Trados also provides a concordance search function which allows users to search the database for instances of a particular word or phrase. This function can also be used to identify potential candidates for partial Iconic Linkage.

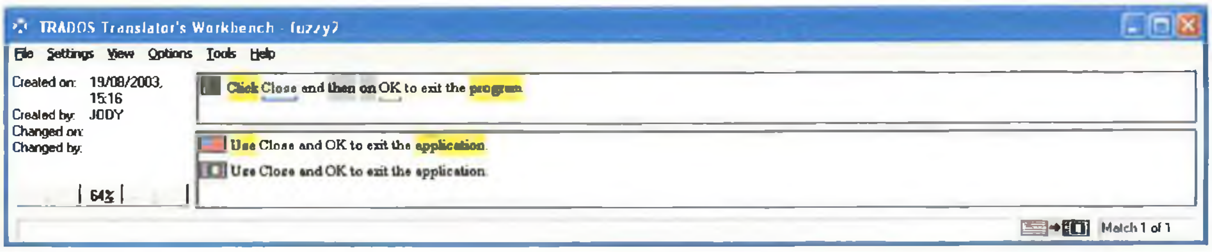


Figure 12: Fuzzy Match Showing Placeables (underlined) in Trados

The primary benefit of this method is that CAT tools can "remember" and analyse a greater number of sentences than a human could ever hope to do. In doing so, CAT tools capitalise on latent Iconic Linkage, be it full or partial. However, it should be noted that this method on its own can only detect latent Iconic Linkage in the text. This can then be reused throughout the text or transformed into full Iconic Linkage depending on the human operator's memory. CAT tools alone cannot introduce new Iconic Linkage into a text, only repeat existing formulations consistently throughout the text.

When style guides and CAT technology are used together, however, they form a powerful suite of methods with each one effectively cancelling out the shortcomings of the other. Thus, style guides can be used to introduce new Iconic Linkage into a text by specifying how something should be phrased while CAT tools ease the burden of analysing and remembering large amounts of text.

In this study, one version of the user guide was rewritten using Trados in conjunction with selected style guide rules. As already stated, it is not feasible to develop a comprehensive style guide specifically for this study so it was decided to use a commercially available style guide published by *Microsoft Press*. The *Microsoft Manual of Style for Technical Publications Version 3.0* was chosen because of its comprehensiveness and because the ubiquity of Microsoft products makes its writing style more familiar to users. It would be unrealistic to implement every rule, indeed many of the rules contained in the style guide simply do not apply to this user guide. Rather, the following series of rules were selected on the basis of how easily they could be implemented and their applicability to DigiLog.

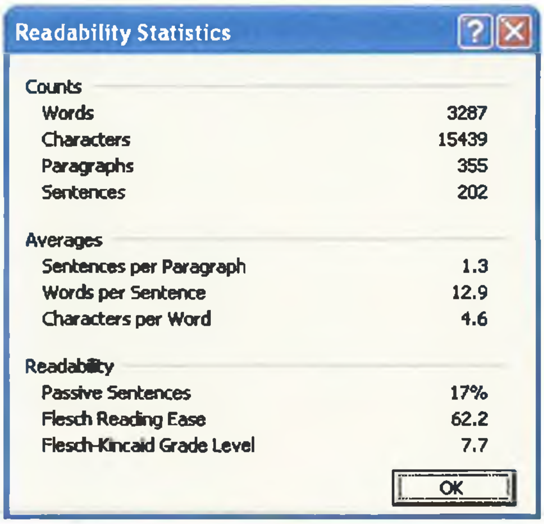


Figure 13: Readability Statistics for Original User Guide

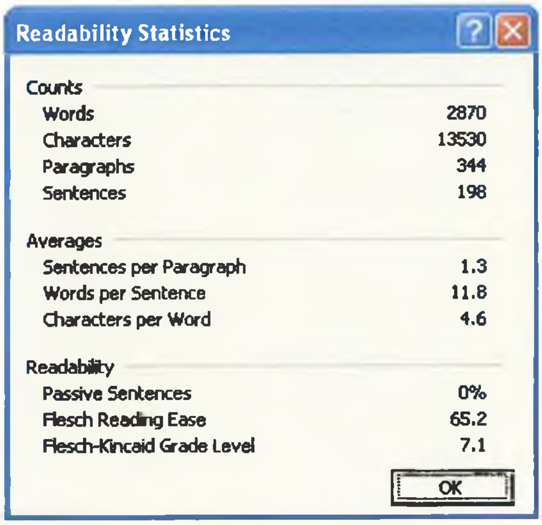


Figure 14: Readability Statistics for Edited User Guide

The readability data also shows that, in addition to reducing the overall word count by approximately 12.7% from 3,287 words to 2,870 words, there are no passive sentences in the rewritten user guide. This is an obvious result of the rewriting process when we consider that one of the rules selected to implement Iconic Linkage explicitly states that the active voice should be used instead of the passive voice.

Nevertheless, it is possible that, even before we conduct the experiment, the elimination of passive sentences could be regarded as a confounding factor. This may or may not be a valid proposition. One argument to support this may be that eliminating passive sentences merely makes the text more readable, not usable. However, referring back to the discussion of readability in Chapter 2, it is apparent that readability is just *one factor* which contributes to usability, and as such is not distinct from it. Thus, any improvements in readability (which, in this case are negligible) are as a result of an attempt to improve usability, i.e., the selected strategies aimed at implementing IL.

In any case, if the results of the experiment do show that users using the rewritten version of the users guide perform better, the nature of the improvement in performance will indicate whether eliminating passives is a genuine confounding variable. If, for example, participants in the experimental group only perform better in terms of the speed with which they work, then it is possible that the missing passives are a confounding variable because we can attribute the increased speed with ease of comprehension caused by improved readability or a shorter text. However, if improvements take place across a range of usability criteria, e.g. speed, error rates, retention of information over time, numbers of errors and satisfaction, then it would be difficult, if not impossible, to attribute such an improvement to improved readability alone. Thus, the issue of whether the elimination of passives can only be answered by the results of the experiment.

These readability scores, while showing that both versions were of a relatively equal standard from a traditional readability point of view, do not show the deeper and more fundamental textual differences between the documents. Instances of full Iconic Linkage can, however, be uncovered using the "Analyse" function in *Trados Translators Workbench.*

Match Type	Original Version	Edited Version
Repetitions - Segments	10	28
Repetitions - Words	21	291
IL Percentage	0	10

Table 18: Comparison of Repetition in Both User Guides

Table 18 contains results from the "Analyse" function run on the original and edited versions of the user guide. While both versions are virtually the same in terms of content and readability, the results clearly show that the amount of repetition or Iconic Linkage in the two versions varies quite significantly. We can see that in comparison to the original version, at least 10% of the total word count of edited version consists of Iconic Linkage as suggested by the fact that 28 out of the 306 segments are repetitions of other segments. (Note: Some partial repetitions could be instances of IL but they are not detected by the analysis tool in Trados).

The finished user guides were proofed one final time before being printed in colour on high-quality paper. The user guides were spiral bound to make them easier to open and place on a desk while reading. *Appendix G* shows the control version of the user guide and *Appendix H* shows the experimental version of the user guide which features Iconic Linkage. *Appendix N* presents a number of examples of the changes made to the Experimental version of the user guide.

Creating a User Profile

In order to select appropriate participants for the usability study, it was necessary to first identify the type of people who would make up the real users of the software being used. For the purposes of creating this profile, a user profile based on that presented by Dumas & Redish (1993:129-133) was developed (see *Appendix A*). The purpose of this was to determine certain characteristics of the users and their backgrounds, skills and qualifications.

This questionnaire was completed by the software manufacturer in consultation with a real user of the software. The questionnaire showed that users are always graduates with a beginner's to intermediate knowledge of computing. None of the users could be defined as experts. The typical user has a background in an area other than politics or computing and all were recruited on the basis of their English language skills. The user profile questionnaire was subsequently used as the basis for selecting participants for the study later on. This will be described below.

Creating an Informed Consent Form

To ensure that all participants in the usability study understood their role in the evaluation process and to ensure that informed consent was obtained, it was necessary to produce a detailed consent form (see *Appendix I*).

This form sets out to explain the nature of subjects' participation in the study and explains what type of data is recorded and how it will be used. To guarantee subjects' welfare and to ensure that they do not feel pressurised, stressed or otherwise influenced, the consent form sets out subjects' rights and entitlements.

Safeguarding Tester Identities

Each participant or tester was assigned a unique identifier known as the *Tester ID*. All documents, files, tapes etc. produced in conjunction with a participant's session were marked with this ID exclusively. In order to keep a record of the participants' details and their IDs, a separate Tester ID sheet was maintained (see *Appendix E*).

Task Sheets

For a usability test to work under the time constraints of a laboratory experiment, the participants must be told what work they should perform using the software. Of course, it is not uncommon for software companies to release advanced prototypes of applications (known as *beta versions*) to a group of users who are simply asked to find errors or problems (Schneiderman 1998:131). However, such an approach, apart from being extremely difficult to standardise, would be impossible to use within the time constraints of a laboratory-based study. Also, open-type "bug hunts" such as this are generally performed by advanced users with more expertise than the participants in our study.

Instead, participants need to be guided through those tasks which will highlight potential usability problems. According to Dumas & Redish (1993:160-163), there are a number of ways of selecting tasks for a usability study. The first way is to use tasks suggested by designers. The authors argue that because designers know the system intimately, they will have a good idea of where potential problems lie. However, it is precisely this detailed knowledge that can result in designers focussing on more complex and advanced problems while overlooking the more lower level problems likely to affect non-expert users.

The second way, which was adopted here, involves basing tasks on what real users will actually do with the software. So, for instance, users of DigiLog would create a new log, change the autosave settings, format the text, save the logs as well as actually logging a debate or meeting. An interesting aspect of choosing tasks is determining their size and scope. If the task is too short or compact, it may take just seconds to complete and will be very difficult to observe and quantify. Conversely, if a task is too long or involves more than one system concept, it may be too taxing for participants. It may also prove difficult to quantify because of the difficulty detecting where one task or subtask ends and the next one starts. In designing the tasks for this study, tasks were chosen that corresponded to a single concept and which were as self-contained as possible.

Here, the issue arises of exactly how self-contained or independent tasks should be and whether they need to be performed in order. Wixon & Wilson (1997:670) define two ways of presenting tasks: *results-based* and *process-based.*

Results-based tasks simply instruct testers to achieve a specific goal without providing any indication of the intermediate stages. Process-based tasks detail the various steps involved in completing a task. For the purposes of this study we are interested in finding out exactly what happens when users perform tasks, not just whether they complete the task or not. As such, a process-based approach was adopted.

With regard to the question of whether process-based tasks should be independent or interdependent, Wixon & Wilson (*ibid.*) state that independent tasks allow participants to move on to the next task regardless of whether the previous task was completed successfully. This contrasts with interdependent tasks where if participants encounter serious problems, they may not be able to proceed on to the next task without some form of intervention from the test administrator.

In reality, however, tasks performed in a working environment are rarely independent, i.e. they are performed as part of a user's strategy for achieving some goal. Frequently, therefore, if users cannot complete a task or subtask, they will not be able to proceed or achieve their goals. This presents us with a compromise between convenience in the usability laboratory and realistic tasks which reflect actual usage scenarios. In this study, it was felt that the nature and length of the tasks would not pose problems which could not be resolved with a minimum amount of interaction with the test administrator, if necessary. Furthermore, the need for realism was deemed to be of greater importance than convenient test administration. For this reason, all tasks were dependent on the completion of the preceding tasks. The tasksheet used in the study is presented in *Appendix B*.

Recorded Speech

As one of the tasks in the study involved logging a speech, it was necessary to source a recording of a speech. Ordinarily, the DigiLog application would be used in conjunction with its sister application DigiTake to play pre-recorded audio files to which the log files would be appended. However, it was not possible to set up the client-server hardware needed to do this due to space and hardware considerations in the test venue. Instead, it was decided to record a speech and play it during the task independently of DigiLog using another playback device.

As DigiLog is generally used in a political environment, it was felt that the speech to be logged should come from this domain. A range of speeches were examined before one featuring just two speakers - a male and a female – was selected (see *Appendix C*). A speech featuring a male and a female speaker was chosen to make it easier for the test participants (who are inexperienced in audio transcription) to differentiate between speakers.

Having selected the text, it was read out by a male and a female actor and recorded digitally. This was done using *Cakewalk Pro 8.0* digital recording software on a PC to which a microphone had been connected through a 16 channel mixing desk. The recorded conversation was normalised and saved in .WAV format. It was then converted into MP3 format using *MusicMatch 7.1* to reduce the file size and saved on CD. The file was played back on a separate PC to the one being used by participants using *Winamp* audio player and the PC's internal speaker.

Task Event Log

The quantitative data collected from the usability study consists of times and the number of occurrences of certain events. In order to collect this information, an event log consisting of two sections was created. The first section simply provided boxes for recording the times needed to perform each task as well as times for several other activities.

The second section consists of a table listing each measurement criterion. Alongside each criterion, there is a list of numbers from 1 to 30 which is used to record each individual instance of an event. As each instance is observed, another number is crossed out. The total at the end of the test is then recorded. A vertical line is drawn after each task to allow a breakdown of instances for each task. *Appendix D* shows the Task Event Log.

Post-Task User Satisfaction Survey

As was discussed in Section 5.2.2.2, usability is measured not just on the basis of how well and how quickly users work with software, but also whether or not they like using the software. Subjective user assessment is a way of determining user attitudes to software – just because users **can** use software does not mean that they like it and that the software is considered usable. This is important for a number of reasons. In discretionary use scenarios (Usability.net: subjective), i.e. cases where users **choose** to use software, low levels of user satisfaction will prompt users to abandon a particular product in favour of another product. In mandatory use scenarios, i.e. cases where users **must** use the software to do their jobs, low satisfaction leads to absenteeism, high staff turnover and a variety of other complaints from an unhappy workforce (*ibid.*).

For the purposes of this study, a modified version of QUIS (see *Appendix J*) was used which omitted the questions relating to *system capabilities*, *multimedia*, *voice recognition*, *virtual environments*, *Internet access* and *software installation*. A number of questions from other sections were also deleted because they were not deemed to be applicable to this study. In addition, certain sections and questions were reworded to make them more specific to this study. A number of questions were added in order to explicitly test the hypothesis of the study. Such questions included:

- Language used in the user guide is consistent
- Language used in the user guide is repetitive

These questions sought to determine whether inconsistency or repetition were negative factors which affected the levels of satisfaction among users (i.e., users may become irritated by excessively repetitive text or fatigued/confused by inconsistent formulations).

Post-Test Questions

As discussed in Section 4.3.1 one of the assessment criteria with which usability is measured in this study relates to how well users remember information over time. Of course, the ideal method for testing retention of information over time is to ask users to use the system again using slightly different tasks which draw on the same information. However, the expense and difficulty in securing participation for an additional test session proved to be prohibitive in this study.

Instead, a variation of the interviewer-administered questionnaire (see page 167) was used. Based on the core knowledge required to perform the test tasks, a series of 15 questions (13 of which were multiple choice) was compiled to test how much users remembered after completing the tasks. The multiple-choice format was chosen to facilitate "cueing" or recognition of answers (see Section 3.4.4). While this may seem erroneous in that it "gives" users the answers, it emulates the type of prompting provided by visual clues on-screen in the software such as menu options, icons, dialog boxes etc. Unfortunately, time constraints mean that we cannot determine which answers were "guessed" or triggered by recognition and which ones the testers actually knew without any cueing. Nevertheless, both types of answer combine to show how much information is retained over time.

5.4.1.2 Test Environment & Tools

The pilot study was conducted in a large and spacious office in the *Centre for Translation & Textual Studies* at Dublin City University. The room was free from clutter, bright and well ventilated so as to provide as pleasant and natural a working environment as possible. Refreshments were provided for participants. The layout of the test equipment and the work area is shown in Figure 15.

The test was conducted on a new PC with a Pentium III 2.4GHz processor and 256MB of RAM, a CD-R drive and a 40Gb hard disk. A video camera was positioned to the right of the participant at a distance which allowed the camera to be as unobtrusive as possible and to record the participant and the surrounding desk area. The aim of this was to record not just the participant's face but also the user guide while it was being used. This was deemed important in order to record the length of time the user guide was used **and** the way in which it was used (i.e. leafing through the pages or carefully reading each page).

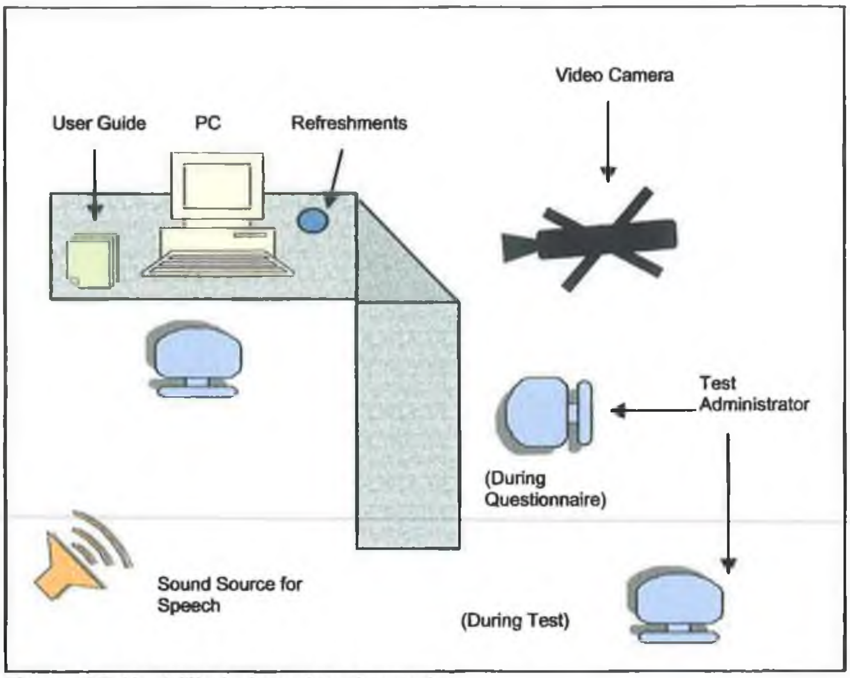


Figure 15: Usability Laboratory Layout

It is also clear from the diagram that the test administrator remained in the room during the test. Although the benefits of indirect observation are clear (see page 159), it was felt that a form of direct observation was necessary for a number of reasons. Firstly, in the absence of a separate observation room with viewing and intercom facilities (see Schneiderman 1998:128) communication with the participant would have been extremely difficult if not impossible. Secondly, having the test administrator in the same room as the participant made it possible for the test administrator to intervene and provide assistance if necessary. This is particularly useful when using interdependent tasks as is the case in this study (as discussed on page 201). It was, therefore, much easier to administer the test.

Two items of software were specially installed on the PC: the test software – DigiLog (see page 192) and the screen logging software (Camtasia).

Screen Logging Software

While user satisfaction questionnaires and video recording go some way to helping us understand the relationship between users and a system, they only provide part of the overall picture. To fully understand users' performance as they carry out tasks, it is necessary to see what they are doing on-screen, i.e. to observe how they actually interact with the system.

Obviously, it would be ill-advised to sit beside the participants and look over their shoulders as they work, not least because it may affect their performance. It would also be physically impractical and would leave us with no permanent record of events. Some solutions proposed involve positioning a video camera in such a way that it can record events taking place on-screen (Preece 1994:619). Other solutions involve connecting an additional monitor to the computer being used by the participant. Known as a slave monitor, it is usually located in another room and shows what is happening on the participant's screen. This approach can be combined with a video camera to provide a permanent record of events. Neither method is particularly attractive because of the additional hardware and technical requirements.

Another approach is to use some form of software logging tool (Preece 1994: 626-627). At their most basic, logging tools can record the keystrokes made by users. More advanced versions also record timestamps for each key that is pressed to give an insight into the time taken to perform tasks. Unfortunately, such models only record actions carried out using the keyboard – they do not record actions carried out using the mouse, e.g. opening menus, selecting items, highlighting text or clicking icons. At the opposite end of the spectrum there are sophisticated tools which take the benefits of interaction logging and video recording.

One such product is *Morae* by Techsmith. This product allows usability researchers to produced synchronised screen and video recordings, observe and mark critical moments and analyze, edit and share recordings. The software can also be used to record audio produced, for example, during think-aloud protocols. As regards user activities, Morae can be used to record changes to web pages, mouse clicks, keyboard input and, using the Remote Viewer, on-screen text and markers. Using such a product would undoubtedly be of great benefit, however, the cost of such a product (USD$1103 at time of writing) is prohibitively expensive for use in this study.

A more feasible approach is to use stand-alone screen-recording software to record real-time moving images of the events taking place on-screen. Generally, such tools are used to create online training tutorials but they can just as easily be used for the purposes of a usability study.

A number of products such as *My Screen Recorder*, *Matchware*, *CamtasiaStudio* and *HyperCam* were evaluated. Of these, only *CamtasiaStudio* proved usable because, apart from its superior functionality, the image quality achieved by the other products was too low to be able to clearly interpret events or even read text.

Camtasia Studio by Techsmith is a suite of tools for recording screen events as well as editing and producing "movies". These tools make it possible to add annotations, audio files, text etc. to recordings. The suite also includes a proprietary player which is used for showing recordings (the recordings can also be played using Microsoft Media Player although the image quality is much better using the Camtasia player).

Recordings are stored in AVI format and with standard compression levels produce perfect quality files; average file sizes are approximately 1MB per minute of recording. There is also a "pack and go" facility which allows high-quality playback of recordings without the need to install Camtasia. The recording tool can be used to specify the precise area of the screen to be recorded – any section of the screen from a dialog box to a window to the entire visible screen area can be selected. Once Camtasia was installed on the PC, separate folders for storing the recordings were created for each participant.

The benefit of using Camtasia is that it records everything that takes place on-screen – even when text is entered. Camtasia cannot, however, record when function keys or keyboard shortcuts are used – it only records the results of such actions provided they have a visual consequence which is displayed on the screen. However, because DigiLog is a mouse and menu-driven application, this is not a problem. What is more, Camtasia records all actions performed on a PC, not just those carried out in DigiLog. Thus, when users switch between applications or windows or when they use the *Start* menu, Camtasia will record it.

Also, because Camtasia features an elapsed time counter, it is possible to calculate the time spent performing tasks and subtasks, simply by measuring the time between the start of the task (or when the mouse first moves) and the end of the task (when the pointer stops). Each task can be labelled using the effects tool, e.g. a banner can be added to the recording for the duration of a particular task.

5.4.1.3 Participants

According to Nielsen (2001) and Dumas & Redish (1993:128), a typical usability study will generally feature a maximum of two groups of five people. This presented a problem for the pilot study in that the numbers required are almost the same as for the main study. The minimum realistic number of participants for a group is three to ensure that the results are reasonably reliable and not the result of the idiosyncratic behaviour of one or two participants (Dumas & Redish 1993:128).

With this in mind, two groups of three participants were recruited for the study. Based on the results of the initial user profile (see page 200), it was decided that participants should be graduates, have excellent English language skills and reasonable PC skills. It was not necessary to look beyond the local campus population for participants because such a group was readily available in the form of students on the *Graduate Diploma/MA in Translation Studies* at Dublin City University. In addition to being a postgraduate course, the course also provides thorough training in both general and translation-related computer applications as well as various English language courses.

An email was sent to all students on the programme explaining the nature of the usability study. Students were asked to participate as co-testers in order to help assess the usability of a software product and its user guide. It was emphasised that participants would not be tested, but rather that they were testing a product and giving their opinions on it. As such, there was no need to feel pressurised, concerned or reticent about their performance. Potential candidates were informed that their involvement would require a commitment of three hours over three weeks for which they would be paid €15 per hour.

One stipulation was made in that participants must be native speakers of English. This was necessary to rule out any potential problems caused by various levels of English language skills among the fairly high proportion of foreign students studying on the programme.

Of the nine respondents, six were chosen at random to take part in the study. The others were thanked for their interest and it was explained that the quota had been reached. Each participant was contacted individually and none were made aware of the identity of the other participants. In an effort to minimise the risk of participants discussing the experiments among themselves, the tests were conducted outside term time so that students would not be in as regular contact with each other. In addition, test sessions were scheduled as far apart as possible to rule out participants meeting in or outside the laboratory. Participants were explicitly requested not to discuss the nature of the experiment; this was reiterated in the terms of the Consent Form (see *Appendix I*). Test sessions were arranged with each participant and recorded in the Tester ID form (see *Appendix E*).

5.4.1.4 Session 1: Familiarisation

The pilot study was conducted in three sessions over the course of three weeks for a total of 2.5 hours. Session 1 was the "Familiarisation" stage of the study and it involved introducing participants to the *DigiTake* package. A product brochure detailing the components of the suite as well as their functions and uses was emailed to each participant. Participants were instructed to spend approximately one hour reading this document. The purpose of this was to familiarise participants with the system so that they would have a context for the information they would learn from the user guide (Foss *et al.* 1981:334). This document aimed to help users understand the software and the tasks they would eventually perform. This document did not, however, provide any specific information on how to use the system or its functions. It simply described the general technology, functions and architecture of the system along with its working environment.

In parallel with this, each participant was contacted to arrange and confirm their test times and dates for Session 2.

5.4.1.5 Session 2: Testing

Session 2 involved conducting the actual test sessions with participants. Upon arrival, each participant was welcomed and offered refreshments. Having been shown to the workstation, the purpose and nature of the test was again explained. It was emphasised that the purpose of the study was to assess the user guide and not the participants' abilities or skills as regards computers, typing, intelligence etc. The participants were told that they were co-testers – they were part of the test team and that they were doing the testing, not being tested. They were also told that they could take a break or withdraw from the study at any time.

Next, each user was given a consent form (see *Appendix I*). They were asked to read it and ask any questions they had. If they were happy with the terms of the consent form, they were asked to sign the consent form. The form was also signed by the test administrator.

At this point, the video camera was started. While the tasks had not yet started, it was felt that turning on the camera at this stage would help participants become accustomed to the camera and the low level of background noise it created before they actually started working. This was intended to minimise any adverse effects caused by the camera's presence.

Once the camera was started, participants were given a randomly pre-assigned user guide. They were told that they had up to 30 minutes to read the user guide. They were also told that they would be allowed to use the user guide during the tasks. Participants were told that during the test, they should not ask questions or otherwise ask the test administrator for assistance unless it was absolutely essential. At all times, they must consult the user guide for answers. If a participant asked a general question, the test administrator referred the participant to the relevant section of the user guide. The test administrator only provided specific information where the question arose as a result of a technical problem with the hardware or software or where, after consulting the user guide the participant was in danger of not completing the task. When participants had finished reading the user guide, a task sheet (see Table 19) was distributed. Participants were informed that the test would proceed one task at a time and that they must not start a new task until told to do so. Upon completion of each task, the participants were asked to indicate this to the test administrator.

Before starting the first task, the test administrator started the Camtasia screen recorder. This simply involved having Camtasia running in the background before the participants arrived and then pressing a single function key on the keyboard to start recording. The first task was explained orally and participants were directed to the task sheet and instructed to commence the task. Upon completion, the participants informed the test administrator and were given the opportunity to ask questions, give comments or take a break. Each task was conducted in this manner.

When all of the tasks had been completed, Camtasia was stopped and participants were again given the opportunity to take a break. Participants were asked whether they had any initial comments on the test. The administrator did not discuss these comments because the only reason for asking this question was to help put participants in an analytical frame of mind whereby they look back on the tasks from the point of view of a tester. This served to prime them for the QUIS questionnaire which was administered next.

In administering the QUIS questionnaire, the test administrator moved to a seat beside the video camera and directly opposite the participant at the desk (see Figure 15 on page 206). Participants were told that a core part of usability testing was to find out how users feel about using a product. They were told that just because something is easy to use, it is not necessarily usable if users do not like using it. Participants were reminded that there were no right or wrong answers. The participants were given the questionnaire and asked to complete it. If any questions were unclear or confusing, participants were told to ask the test administrator.

Participants were then thanked for their assistance and the importance of their participation was emphasised. At this point, the video camera was turned off. During the first and second sessions, however, the camera was turned off immediately after the QUIS questionnaire was administered (see *Appendix J*). It soon became apparent, however, that informal comments which sometimes continued right up to the point the participant left the room were not recorded. For this reason, the camera was left on until each remaining participant left the laboratory.

5.4.1.6 Session 3: Post Test Survey

Session 3 involved administering the post test survey. These sessions were carried out exactly one week after each participant's second session. The post-test survey involved administering the multiple-choice test sheet (see *Appendix F*) and took approximately ten minutes. Participants were given the opportunity to ask questions about the study as a whole. Participants were thanked for their participation which was now at an end.

5.4.1.7 Findings of Pilot Study

The results of the pilot study can be grouped into the following categories:

- Results of Time Measurements
- Results of QUIS Questionnaire
- Error Rates
- Implications for Experimental Design

Results of Time Measurements

In calculating the times for the various activities set out in the Task Event Log (see *Appendix D*), it soon became apparent that this list contained criteria and aspects of subjects' performance which could not be detected easily, if at all. Problems arose mainly because of the difficulty in establishing exactly what a subject was doing at a given point in time. Thus, it was not always possible to distinguish between a subject who was recovering from an error and a user who was performing a task normally.

Similarly, determining when users could be regarded as being unproductive was impossible when we consider that users may be actively thinking about a problem or looking for information on screen while giving the impression of inactivity. One way of combating this would be to implement some form of think-aloud protocol but for reasons described already (see the discussions of Audio Recording on page 159 and Verbal Protocolson page 162) it was felt that the use of think-aloud protocols would be of limited use here.

Another problem which arose in the study was that of different subjects having different levels of manual dexterity. The result of this was that measuring the time subjects spent navigating menus was not realistic quite simply because it took some subjects longer to physically make their way through the menus. It would seem more appropriate to treat menu navigation as discrete events whereby each occurrence of a subject search through a number of menus apparently looking for the desired option is counted on a "per incident" basis.

Ultimately, the only useful and feasible measurement which could be carried out was the time taken to complete each task. Table 19 lists the nature of each individual task.

Task 1	Create two new entries in *QuicKey*
Task 2	Create a new log in *DigiLog* and configure the automatic save settings and set the working directory.
Task 3	Logging task
Task 4	Format the text in the log
Task 5	Manually save the log in RTF format to a specific location

Table 19: Pilot Study Tasks

It was not possible to record times for subtasks because of the difficulty in establishing precisely when one subtask had been fully completed. Some subjects returned to certain parts of tasks which confounded matters. It should also be remembered that Task 3 involved subjects logging a pre-recorded speech. As such, this task always lasted for the same duration (i.e. 5:03 minutes). Consequently, there was little sense in including the times for this task as the fixed time tells us nothing about a subject's performance. This particular task is better used as an opportunity to examine the usability criteria exclusively.

	Control Group				
	T1	**T2**	**T4**	**T5**	**Totals**
P1	982s	124s	177s	74s	1357s
P3	848s	417s	111s	191s	1567s
P5	720s	403s	411s	72s	1606s
Group Averages	850s	314.67s	233s	112.33s	1510s

Table 20: Task Times for Pilot Control Group

In the case of the Experimental group, problems arose with subject P4 who failed to follow instructions regarding reading both the familiarisation material sent out as part of Session 1 and with regard to reading the user guide for the specified length of time during Session 2. Consequently, the subject was unable to complete the tasks and as such, no data is available for this particular subject. Table 21 shows the task times for the other two participants in the Experimental group.

	Experimental Group				
	T1	T2	T4	T5	Totals
P2	375s	77s	93s	90s	635s
P4	--	--	--	--	--
P6	277s	363	70s	271s	981s
Group Averages	652s	440s	163s	361s	808s

Table 21: Task Times for Pilot Experimental Group

From the figures presented above and presented in Figure 16, there appears to be a clear improvement in the tasks times for the Experimental groups in all tasks except Task 5. However, because the data is incomplete as a result of P4, it is impossible to use it as a basis to draw any reliable conclusions as to the effect of Iconic Linkage on usability.

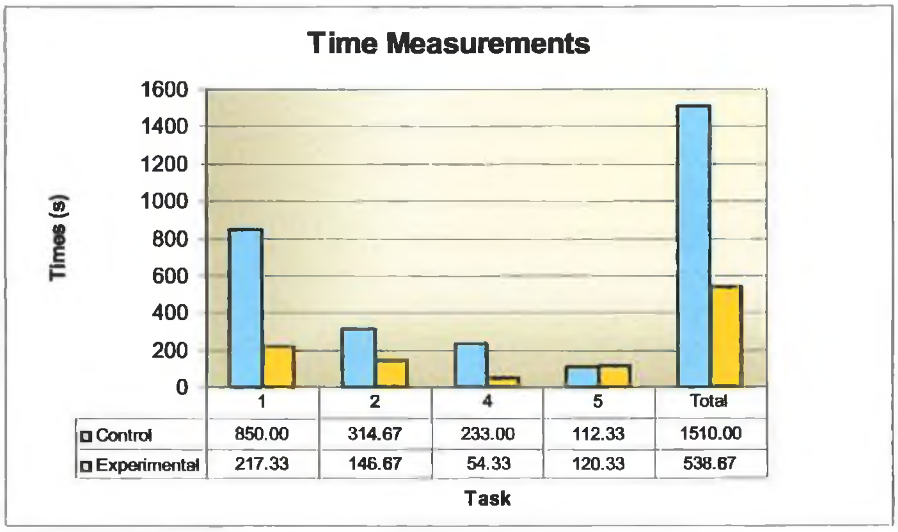


Figure 16: Graph Showing Overall Group Times for Pilot Study

Error Rates

Of the large list of criteria on page 177, only some proved to be useful in terms of detecting errors during the test. The following tables illustrate only those error criteria which provided at least one observable instance during the test. The remaining criteria have been omitted. In gathering the data, the scores from the post-test survey were used as data for the criterion "Commands & Icons Remembered Afterwards".

In the Table 22 and Table 23, this data is referred to as the "PTS Score". Once the data has been compiled, all of the figures except the PTS Score and the number of tasks completed were added together. The PTS Score and number of tasks completed were subtracted from this total to give an overall error score. The purpose of this score is to facilitate comparing the number of errors made by each subject.

	P1	P3	P5	Group Total	Average
Tasks Completed	-4	-2	-2	-8	-2.66
Times UG Used	12	9	15	36	12
PTS Score	-8	-7.5	-10	-25.5	-8.5
Incorrect Icons	2	6	3	11	3.66
Incorrect Menus	7	6	8	21	7
Verbal Interactions	7	6	5	18	6
Confusion	8	5	7	20	6.66
Satisfaction	0	0	0	0	0
Error Score	24	23.5	26	72.5	24.16

Table 22: Usability Criteria Assessment & Error Scores for Pilot Control Group

	P2	P4	P6	Group Total	Average
Tasks Completed	-5	--	-4	-9	-4.5
Times UG Used	0	--	6	6	3
PTS Score	-11.5	--	-9	-20.5	-10.25
Incorrect Icons	2	--	1	3	1.5
Incorrect Menus	2	--	3	5	2.5
Verbal Interactions	0	--	1	1	0.5
Confusion	1	--	1	2	1
Satisfaction	1	--	0	1	0.5
Error Score	-10.5	--	-1	-11.5	-5.75

Table 23: Usability Criteria Assessment & Error Scores for Pilot Experimental Group

Table 23 shows the data collected from the Experimental group. Again, owing to the difficulties encountered with participant P4, it was impossible to gather data for this part of the experiment. Nevertheless, when calculating average scores for the Experimental group, P4 was treated as a non-scoring member of the group so that averages could be calculated on the basis of equal group numbers.

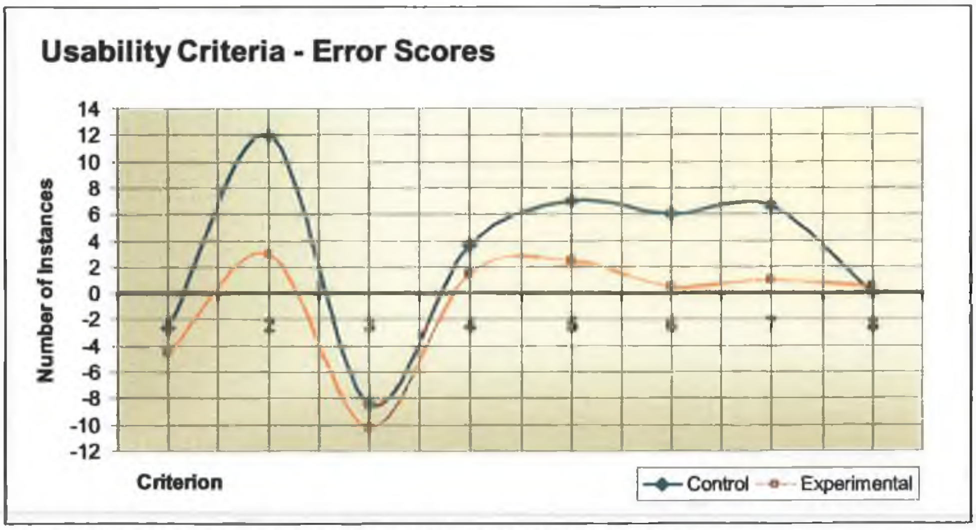


Figure 17: Initial Graph Attempting to Show Pilot Error Scores

However, it became apparent when presenting this data in the form of a graph, that there were problems in the way the results were calculated for each criterion. Figure 17 shows that the results for the "Tasks Completed" (Criterion 1) and particularly the PTS Score (Criterion 3) noticeably skew the graph. This fact highlighted faulty logic in the way criteria were worded and applied. The list of usability criteria provided a list of errors which each participant committed. As such, it makes little sense to subtract the number of correct answers given from the total number of errors. Instead, it was decided to define the PTS Score as the number of incorrect answers provided in the post-test survey as this is more compatible with the aim of the remaining criteria. The criterion governing the number of tasks completed was reworded as "the number of tasks *not* completed". Figure 18 illustrates the graph with the new data.

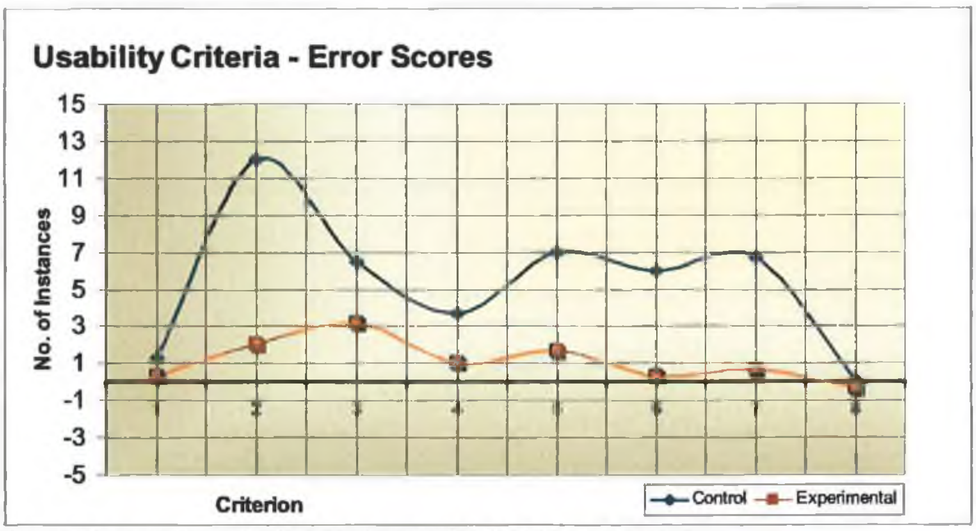


Figure 18: Usability Criteria Error Scores for Pilot Study Groups

Table 19 shows the average overall error rates for each group in the form of a bar chart based on the modified method for applying criteria. It shows, even taking into account the effect of what were effectively a series of zeros for P4, that the Experimental group appears to have performed much better than the Control group. However, because the data is incomplete, it is impossible to use it as a basis to draw any reliable conclusions as to the effect of Iconic Linkage on usability.

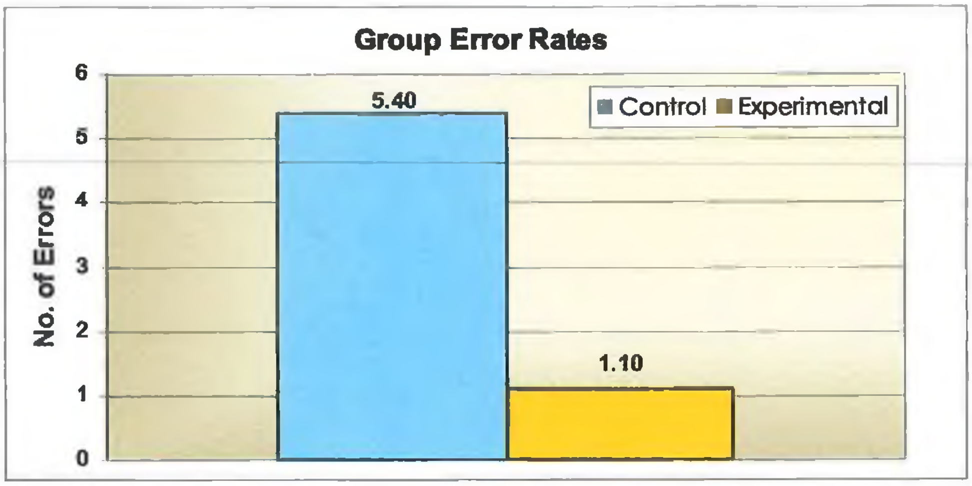


Figure 19: Average Group Error Rates - Pilot Study

Results of QUIS Questionnaire

Table 24 presents the average ratings for the Control and the Experimental group in the pilot study. These figures are presented in the form of a graph in Figure 20.

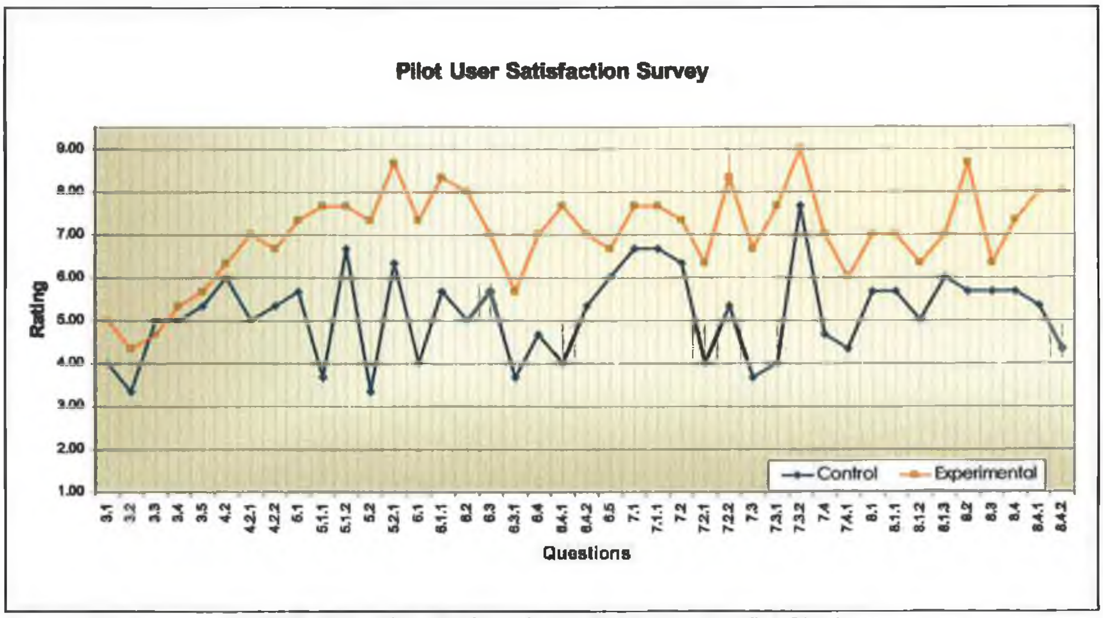


Figure 20: Average QUIS Satisfaction Ratings for Both Groups in Pilot Study

Question	Control	Experimental
3.1	4.00	5.00
3.2	3.33	4.33
3.3	5.00	4.67
3.4	5.00	5.33
3.5	5.33	5.67
4.2	6.00	6.33
4.2.1	5.00	7.00
4.2.2	5.33	6.67
5.1	5.67	7.33
5.1.1	3.67	7.67
5.1.2	6.67	7.67
5.2	3.33	7.33
5.2.1	6.33	8.67
6.1	4.00	7.33
6.1.1	5.67	8.33
6.2	5.00	8.00
6.3	5.67	7.00
6.3.1	3.67	5.67
6.4	4.67	7.00
6.4.1	4.00	7.67
6.4.2	5.33	7.00
6.5	6.00	6.67
7.1	6.67	7.67
7.1.1	6.67	7.67
7.2	6.33	7.33
7.2.1	4.00	6.33
7.2.2	5.33	8.33
7.3	3.67	6.67
7.3.1	4.00	7.67
7.3.2	7.67	9.00
7.4	4.67	7.00
7.4.1	4.33	6.00
8.1	5.67	7.00
8.1.1	5.67	7.00
8.1.2	5.00	6.33
8.1.3	6.00	7.00
8.2	5.67	8.67
8.3	5.67	6.33
8.4	5.67	7.33
8.4.1	5.33	8.00
8.4.2	4.33	8.00
Group Averages	**5.14**	**7.02**

Table 24: Average QUIS Results for Pilot Study

Appendix J presents the full list of questions used in this questionnaire while *Appendix L* presents the individual ratings provided by each participant in the two groups. It should be pointed out that the data for participant P4 is included in this analysis of QUIS satisfaction ratings. While it may seem unusual to include data from P4 given the problems encountered with this participant during the experiment, this is not as problematic as it first appears. In the previous sections, a series of null values were included for participant P4 so as to ensure the same number of participants in each group. Furthermore, Frøkjær *et al.* (2000) have shown that there is no direct correlation between efficiency, speed and satisfaction and so it is permissible to use data from one without affecting the others. In any case, the participant did use the software and to a certain extent, the user guide and so was fully qualified to provide satisfaction ratings. The overall satisfaction ratings obtained in the pilot study show a marked difference between the two groups.

This is also apparent in relation to individual questions with the exception of question 3.3 ("Overall reactions to the system: dull/stimulating"). This particular response appears quite incongruous with the other questions in Section 3 of the QUIS questionnaire which ask participants to rate the system as "terrible/wonderful", "frustrating/satisfying", "difficult/easy" and "rigid/flexible". This result could be regarded as an anomaly, particularly in light of the responses to questions 3.1, 3.2, 3.4 and 3.5 as the overall response seems to be elevated by a single rating of 6 in the Control group (participant P5). Another possible explanation might be that the nature of the tasks, which admittedly were not particularly stimulating, influenced participants ratings.

5.4.1.8 Implications for Experimental Design

Over the course of the pilot study and during the subsequent analysis of the data, it became apparent that there were a number of areas relating to the design of the experiment that were either unsuitable, insufficiently refined or which were otherwise problematic. The problems encountered can be grouped under the broad categories of: *Tasks*, *Data Collection* and *Criteria*.

Tasks

There were a number of issues relating to the tasks to be performed as part of the test which gave rise to certain problems. The first problem was in relation to the manner in which subjects were prepared for the tasks. Central to the entire test was the need for all participants to read the user guide for 30 minutes. It emerged during the tests that 30 minutes was too long a period of time. Having noted the average times spent by each subject, a period of 20 minutes emerged as a more realistic period of time.

Another problem with the length of time subjects should spend reading the user guide emerged during the session with subject P4. This participant failed to read the brochure provided as part of Session 1 and despite numerous requests, this subject refused to read the user guide for the required period of time, arguing that this was not how the participant normally used a user guide. The participant was asked a number of times to read the user guide but became agitated and defensive. Consequently, the participant did not read the guide and so was unable to complete any of the tasks. It is clear from this that it is absolutely essential that participants comply with all instructions and that the there needs to be some way of firmly insisting that participants read the user guide for the full period of time. It is proposed that participants be told quite openly from the outset that they must spend the full time reading the user guide and that this is to ensure consistency and the effectiveness of the experiment.

Problems also arose with regard to Task 5 which involved subjects formatting the text in the log created as part of Task 3. It was clear that all subjects had a good level of knowledge of *Microsoft Word* and that this interfered with the solutions subjects chose as part of Task 5. Although there are two ways of formatting text, i.e. using the tool bar or the menus, each subject without exception used the tool bar which used the same icons as *Microsoft Word*. So rather than using the information in the user guide, subjects were using their existing knowledge of another application to complete the task. It was decided, therefore, to omit this task as it did not provide any meaningful insight into the usability of the user guide.

Nevertheless, the task did show that both groups were well matched in terms of computer skills so it indicated that the method for selecting participants was effective.

Data Collection

During the test sessions, a number of areas became apparent where the method of collecting data could be improved. The first such area involved keeping the video camera running until the subjects had left the laboratory. The reason for this was that some subjects continued giving feedback right up until they left the lab. When the video camera had been switched off immediately after the post-task questionnaire, these comments were lost. However, during the analysis of the questionnaire data, it soon emerged that the comments and feedback elicited from subjects, while very interesting from a software and document design point of view, were not particularly interesting from the point of view of assessing the usability of the user guide. As a result, it was decided that this type of information should not be collected during the main study.

It also emerged that *Camtasia* was not suitable for recording entire test times because it was not possible to see what was going on when a subject appeared to stop working. It was impossible to tell whether the subject was reading the user guide, talking to the administrator or staring at the screen. In addition, some subjects immediately began tackling a task on-screen using the mouse while others immediately began re-reading the user guide. It was decided that the video recordings would be the better for recording times. Task times were recorded from the time subjects were instructed to start to the time the subject announced completion of the task. Furthermore, although Camtasia files can be used to record partial tasks, it is too difficult to do so in practice because some subjects showed a tendency to repeat subtasks. It was decided that Camtasia would still be used to detect incorrect menu and icon choices etc. in the main study, however.

Several subjects experienced difficulties in hearing and understanding the recorded speech used during Task 3. The speech needed to be either re-recorded or digitally enhanced to make it of an acceptable quality.

Some problems were caused by unclear instructions and a misprint in the task sheet and imprecise verbal instructions from the test administrator. Before conducting the main study the task sheet was clarified and updated and verbal instructions to participants were properly scripted. This modification to the experimental method had an additional benefit in that it helped control the amount variability which may inadvertently come as a result of unscripted instructions from the test administrator. For example, on any particular day, the administrator may give more or less information to the participant. Scripting the instructions given by the administrator, coupled with the strictly enforced procedure for dealing with questions (see page 211) helped rule out the possibility of the administrator making *ad hoc* comments which might vary from session to session and consequently biasing or assisting individual participants.

Criteria

From the relatively long list of criteria used to assess usability, it emerged that many were unsuitable for use in this study, either because relevant events did not occur with enough frequency to justify their inclusion or because they were impossible to quantify given the procedures and equipment in use. As a result, some criteria were deleted while others were rephrased slightly to make it easier to quantify them. An example of a rephrased criterion is "Errors at point where subject thinks task completed" which was rephrased as "Number of times subject stops work without completing a task". It was also necessary to add additional criteria as a result of other phenomena which were observed during the test. The pilot study also made it clear that additional work was needed to establish exactly how the criteria should be applied in the main study.

When analysing the error criteria data, it was apparent that the method for handling data from the Post-Test Survey was not appropriate (see page 217). Instead, the number of **incorrect** answers was added to the totals for the other error criteria. Apart from faulty logic, subtracting the number of correct answers from the total errors meant that any attempts to represent the data in the form of a graph produced seriously skewed results.

In addition, it was felt that the application of error criteria was subjective and, at times, inconsistent. This can be attributed to the fact that a single person was responsible for determining whether an incident actually corresponded to one of the criteria. It was decided, therefore, to clearly define each criteria and what each one involved. Table 25 presents the modified list of error criteria and the definition of each criterion which must be observed in order to be recorded.

	Criterion	Definition
1	Tasks Not Completed	The failure of a user to complete a task despite the administrator identifying the precise page in the user guide; ultimately resulting in the administrator giving explicit verbal instructions on how to complete the task.
2	Times user guide used	Each occasion where the participants stops working to read the user guide.
3	PTS Score	The number of questions answered incorrectly by each participant in the post-task survey.
4	Incorrect icon choices	Where a user clicks an icon which is not associated with the task currently being performed.
5	Incorrect menu choices	Where a user chooses a menu option not associated with the task currently being performed OR where a user scrolls through several menus without choosing an option.
6	Verbal Interactions /Questions During Tasks	Each occasion where a participant asks a question relating to the way a task should be performed or whether a task has been completed.
7	Observations of Frustration	Incidents where a participant expresses frustration verbally or where a participants body language (e.g. sighing) or facial expressions (e.g. frowning) indicate frustration.
8	Observations of Confusion	Incidents where a participant expresses confusion verbally or where a participants body language (e.g. head-scratching) or facial expressions indicate frustration.
9	Observations of Satisfaction	Incidents where a participant expresses confusion verbally or where a participants body language or facial expressions indicate frustration (e.g. smiling).
10	Incorrect Commands / Input	Incidents where a participant uses an incorrect shortcut key or types incorrect commands into a field.
11	Stopped Work without Completing Task	Each instance where a participant mistakenly believes the task to be complete or where a participant gives up.

Table 25: Revised Error Criteria & Definitions

5.4.2 Main Study

The main study was conducted four weeks after completion of the pilot study. By this stage all of the necessary changes highlighted in the pilot study had been implemented.

The venue for the main study was a usability laboratory set up in offices in the Houses of the Oireachtas (Irish Parliament) in Dublin where the DigiTake product is in use. This was made possible by *DigiTake Software Systems Ltd.*, the company who manufactures the DigiTake package. The company also facilitated the recruitment of participants. Owing to the fact that Parliament was in session at the time of the main study, access to both the laboratory and subjects was restricted with the result that it was possible to conduct only one session per day. The main testing sessions took ten working days to conduct.

In terms of preparations, the majority of the work involved modifying the task sheets (see *Appendix B*) to reflect the omission of the text formatting task (Task 4 in the pilot study). The task event logs were also updated to reflect the new usability criteria. Table 26 contains the tasks that were retained in the main study:

Task 1	Create two new entries in *QuicKey*
Task 2	Create a new log in *DigiLog* and configure the automatic save settings and set the working directory.
Task 3	Logging task
Task 4	Manually save the log in RTF format to a specific location

Table 26: Main Study Tasks

During the course of the pilot study it also became apparent that the quality of the recorded speech was not entirely satisfactory. Some subjects found the volume to be too low in places and that the occasionally short gap between speakers did not always allow them to finish typing the previous sentence. Rather than re-record the speech again, the speech file was edited using the *Cakewalk 8.0* digital audio package. The volume levels were normalised to enhance sound quality and gaps of 2-3 seconds were inserted between each speaker's turn to allow subjects to complete typing the previous sentence. A set of high-quality speakers were obtained for playing the speech as the sound quality obtained from using the PC's internal speaker was inadequate.

The overall layout of the usability laboratory was kept almost identical to that used in the pilot study although certain physical features of the office geography necessitated minor changes.

5.4.2.1 Tools

As in the pilot study, the workstation consisted of a high-specification PC onto which *DigiLog* and *Camtasia* had been loaded. Again, labelled directories, forms and video cassettes were readied for each subject. A second PC was used to play the recorded speech.

5.4.2.2 Participants

For the purposes of the main study, it was felt that greater realism could be achieved by conducting the test in the same environment as real users would use the software. In this case, the software would be used in the Irish Parliament. It was also felt that because of the venue for the tests, it would be possible to gain access to subjects who more closely reflect the real users of the software.

With the assistance of *DigiTake Software Systems Ltd.*, contact was made with Fianna Fáil, the political party in Government at the time of the study, and permission was obtained to recruit the party's personnel assistants as subjects in the study. This particular group is much more suitable for participation in the study because, in addition to having a very high proportion of graduates with appropriate levels of computer knowledge, potential subjects are working in the same environment as parliamentary reporters and as such demonstrate an excellent understanding of politics and current affairs. As such, by drawing subjects from this group, it was possible to achieve a much higher level of realism.

The same selection criteria were used in the main study as were used in the pilot study: subjects must be graduates, must be native speakers of English with excellent communication skills and beginner to intermediate level computer skills (i.e. 1-3 years experience). A total of 10 suitably qualified participants were selected at random. The details of each participant were recorded in the Tester ID form and each person was assigned a unique ID number (cf. Section 5.2.3.2).

5.4.2.3 Method

The main study was conducted in three stages over four weeks. During the familiarisation stage, the same product brochure was emailed to participants to explain the nature of the product and the context in which it is used. Participants were asked to read the document carefully.

Each participant was scheduled for a test session over the course of the following two weeks. The procedure for conducting test sessions was almost identical to that in the pilot study. However, the changes highlighted in Section 5.4.1.8 were incorporated into the procedures. In addition, as previously mentioned, the text formatting task was omitted from the test.

Upon arrival, each participant was welcomed and offered refreshments. The purpose of the study was again explained and participants were asked to read and sign the consent form. Participants were then given one of the user guides to read for 20 minutes. At the same time, the video camera was started. After the participants had finished reading the user guide, they were given a task sheet. It was explained that they should move on to the next task only when instructed to do so. A verbal explanation of the tasks written on the task sheet was also given to ensure clarity and to give participants the opportunity to ask questions.

Participants were told that during the task, they should always try to find answers and solutions to problems from the user guide. Participants were told that they could only ask questions or ask for help from the test administrator as a last resort. The Camtasia screen recorder was started and participants were instructed to begin when ready.

During the test, where participants asked questions the answers to which were contained in the user guide, the test administrator directed the participant to the relevant page in the user guide. If, after reading the relevant section of the user guide, a participant still encountered difficulties and appeared to be in danger of not completing the task, the test administrator provided more explicit and detailed information; each of these instances counted as a separate interaction with the administrator.

When participants had completed the tasks, Camtasia was stopped and participants were again given the opportunity to take a break. Participants were asked whether they had any initial comments on the test. The administrator did not discuss these comments because the only reason for asking this question was to help put participants in an analytical frame of mind and prime them for the QUIS questionnaire which was administered next (see *Appendix J*). Arrangements were then made to meet each participant exactly one week later in order to complete the post-test survey. The post-test survey was completed in the presence of the test administrator and lasted on average 10 minutes.

5.5 Results of Usability Evaluation

The following pages present the data collected during the empirical study. This data is presented in tabular form and followed in subsequent sections by an analysis and discussion of the results and their implications.

5.5.1 Results of Time Measurements

	Task 1	Task 2	Task 4
P1	340	175	73
P3	355	210	41
P5	345	280	70
P7	352	442	136
P9	350	254	76
P2	225	110	64
P4	199	109	62
P6	208	83	72
P8	232	120	70
P10	214	112	65

Table 27: Individual Task Times (in seconds) for Participants in Main Study

Group Averages	Task 1	Task 2	Task 4	All Tasks
Control	348.4	272.2	79.2	233.3
Experimental	215.6	106.8	66.6	129.7

Table 28: Average Group Times (in seconds) for Each Task in Main Study

Results of Error Rate Assessment

		Control Group					Experimental Group				
	Criterion	**P1**	**P3**	**P5**	**P7**	**P9**	**P2**	**P4**	**P6**	**P8**	**P10**
1	Tasks Not Completed	1	0	1	1	0	0	0	0	0	0
2	Times user guide used	2	4	3	6	3	2	1	2	3	2
3	PTS Score*	9.5	8.0	8.5	9.0	11.5	4.5	4.5	6.5	3.0	4.0
4	Incorrect icon choices	3	2	2	2	4	2	1	0	1	0
5	Incorrect menu choices	5	6	6	10	7	2	1	2	1	3
6	Verbal Interactions /Questions During Tasks	2	1	3	2	1	0	0	0	0	0
7	Observations of Frustration	1	0	1	1	0	0	0	0	0	0
8	Observations of Confusion	1	3	4	3	2	0	1	1	1	0
9	Observations of Satisfaction (subtract)	0	0	0	0	0	0	0	0	-1	0
10	Incorrect Commands / Input	6	1	4	3	3	3	0	1	2	1
11	Stopped Work without Completing Task	2	0	0	0	1	0	0	0	0	0
	Individual Error Scores	**33**	**25**	**33**	**37**	**33**	**14**	**9**	**13**	**10**	**10**

Table 29: Error Criteria Scores for Participants in Main Study

	Criterion	Control	Experimental
1	Tasks Not Completed	0.6	0
2	Times user guide used	3.6	2
3	PTS Score*	9.3	4.5
4	Incorrect icon choices	2.6	0.8
5	Incorrect menu choices	6.8	1.8
6	Verbal Interactions/Questions During Tasks	1.8	0
7	Observations of Frustration	0.6	0
8	Observations of Confusion	2.6	0.6
9	Observations of Satisfaction (subtract)	0	-0.2
10	Incorrect Commands / Input	3.4	1.4
11	Stopped Work without Completing Task	0.6	0
	Group Error Scores	31.9	10.9

Table 30: Average Error Scores for Each Group in Main Study

*** PTS Score = Number of commands, icons & menus NOT remembered in Post-Test Survey**

5.5.3 Average Ratings from QUIS Questionnaire

Question	Control	Experimental
3.1	5.00	6.00
3.2	3.40	6.00
3.3	6.00	6.75
3.4	3.80	5.50
3.5	3.80	5.75
4.1	5.60	6.00
4.11	5.40	5.60
4.12	6.00	6.20
5.1	6.00	7.80
5.1.1	6.00	7.20
5.1.2	6.60	8.00
5.2	6.60	8.00
5.2.1	4.80	7.00
6.1	4.60	6.60
6.1.1	4.20	6.20
6.2	6.20	7.80
6.3	4.20	7.20
6.3.1	5.00	6.00
6.4	5.20	7.00
6.4.1	4.20	7.00
6.4.2	5.40	7.00
6.5	5.00	6.20
7.1	6.20	7.40
7.1.1	5.40	6.20
7.2	6.20	7.00
7.2.1	4.20	6.40
7.2.2	5.80	7.60
7.3	4.00	6.20
7.3.1	5.40	6.80
7.3.2	6.40	8.00
7.4	4.00	7.40
7.4.1	4.20	6.40
8.1	4.60	7.20
8.1.1	5.40	7.20
8.1.2	3.60	6.20
8.1.3	4.00	6.40
8.2	5.00	6.20
8.3	6.20	6.20
8.4	5.40	6.40
8.4.1	5.40	7.20
8.4.2	4.00	6.60
Group Averages	**5.08**	**6.73**

Table 31: Average QUIS Satisfaction Responses from Groups in Main Study

Appendix J presents the full list of questions used in this questionnaire while *Appendix M* presents the individual ratings provided by each participant in the two groups.

5.5.4 Analysis of Results

Having collated and recorded the data gathered during the main study and observed the differences between the Control and Experimental groups, the next step is to ascertain whether such differences are statistically significant or simply due to chance.

The nature of this study and the fact that only 10 subjects were used, poses a number of problems for any attempts at statistical analysis. Generally, the reliability of data is calculated using methods such as a *Student's t-Test*. However, such tests require at least 20-30 observations for the results to be statistically reliable. This means that the data from the main study cannot be analysed using the more common statistical tests.

This problem can be overcome, however, by using a Rank-Sum test, also known as the Wilcoxon-Mann-Whitney exact test. It is used for comparing two populations to detect "shift alternatives". That is, the two distributions have the same general shape, but one of them is shifted relative to the other by a constant amount under the alternative hypothesis (see below). This test can be used for either continuous or ordinal categorical data. It is one of the most popular nonparametric tests for detecting a shift in location between two populations and is "the only form appropriate for comparing groups of sample size 10 or smaller per group" (Helsel & Hirsch 1993:119).

At the simplest level, the Wilcoxon-Mann-Whitney exact test is used to determine whether one group tends to produce larger observations than a second group. The null hypothesis of this test states that the probability of one group (x) producing a higher value than another group (y) is 50%. This can be written as:

$H_{0:}$ *Prob[x>y] = 0.5*

Helsel & Hirsch (*ibid.*) state that the alternative hypothesis is one of the following:

$H_{1:}$ *Prob[x>y] ≠ 0.5 (two-sided test where x might be higher or lower than y)*

$H_{2:}$ *Prob[x>y] > 0.5(x is expected to be larger than y)*

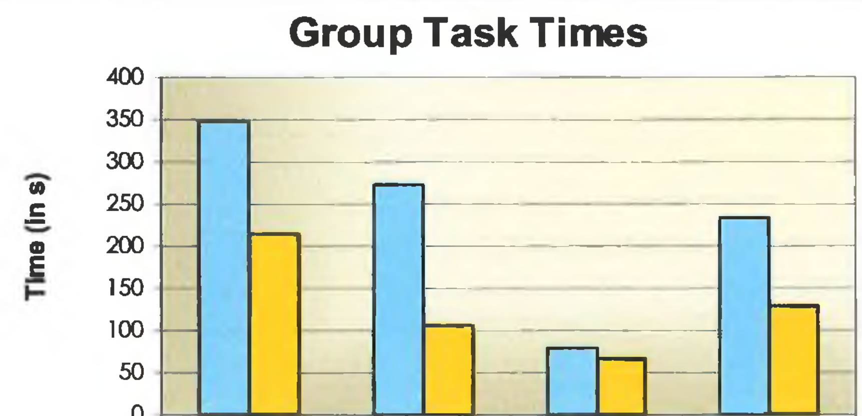
$H_{3:}$ *Prob[x>y] < 0.5(x is expected to be smaller than y)*

Due to the limited sample involved in this study (n=5), the usual asymptotic p-values would not reliable, so exact p-values were calculated throughout using *StatXact 4.01* from Cytel Software Corporation, Cambridge MA.

In performing statistical analyses on the three sets of data, i.e. task times, error rates and user satisfaction levels for each group, it was necessary to decide upon the level of detail to be used. While analysing, for example, data for individual error criteria or times for individual tasks would be interesting, the more detailed analyses we perform the more likely we are to encounter a Type I error (Norman 2003). Type I errors refer to spurious results which can look credible but which are in fact wrong or misleading. This problem is exacerbated by the small sample size used in this study. Consequently, it was decided to analyse the overall data for each subject, i.e. total task times, total error rates and overall satisfaction ratings.

5.5.4.1 Task Times

Total time taken to complete tasks. A two-sided Wilcoxon-Mann-Whitney exact test of the null hypothesis of no difference between total times for the Control group and Experimental group yielded a *p-value of 0.0079* indicating that there is a statistically significant difference between the times for the Control group and the Experimental group.



	Task 1	Task 2	Task 4	All Tasks
Control	348.40	272.20	79.20	233.27
Experimental	215.60	106.80	66.60	129.67

Figure 21: Average Task Times for Each Group in Main Study

From the above graph it is clear that the Experimental group performed the tasks significantly faster than the Control group; the Experimental group were on average 44.4% faster than the Control group. As pointed out before, these figures do not include Task 3 which was the logging task. The reason for this is that the task was of a fixed length, i.e. 302 seconds. As such, including this task would achieve nothing other than inflating the group times of each group by the same amount.

An interesting issue is raised by the results for Task 4. When each task time was analysed individually, two-sided Wilcoxon-Mann-Whitney exact tests yielded p-values of 0.0079 for both Task 1 and Task 2. However, the p-value for Task 4 was 0.254 which indicated that there was no statistically significant difference between the groups. Indeed, Figure 21 clearly shows that while there is a difference between the two groups, the difference is not as pronounced as in the other tasks. There are a number of possible explanations for this. Firstly, it is possible that analysing each task time individually resulted in a Type I error or spurious result.

Another possible explanation is that, bearing in mind this task involved selecting a method for saving a completed log file to a specific location, the task is not as complex as the other tasks and as such there is a smaller likelihood of serious problems arising. Indeed, it could be argued that this task had more to do with users' knowledge of the Windows operating system than their knowledge of DigiLog and as such, this - and not the user guide - played a greater role in the outcome. Nevertheless, analysing the overall time taken by each group to complete all tasks, we can see that the Experimental group performed the tasks faster than the Control group.

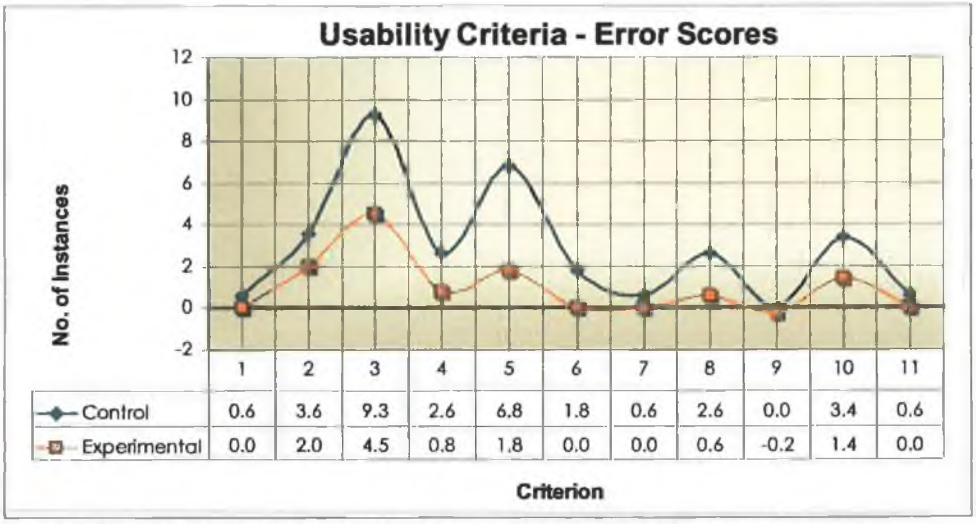
An interesting issue raised by the results for the main study is that the times are significantly lower than those observed in the pilot study. This can be attributed to the fact that, in comparison to the pilot group, the participants in the main study were more experienced in the particular work environment, had a greater understanding of the context in which the software would be used and used computers every day. While the pilot group were a close match to the profile of real users, the participants in the main study were an even better match.

5.5.4.2 Error Rates

A two-sided Wilcoxon-Mann-Whitney exact test of the null hypothesis of no difference between the total error score for the Control group and Experimental group yielded a *p-value of 0.0079* indicating that there is a statistically significant difference between the scores for the Control group and the Experimental group.

From Figure 22 we can see that the Control group using the original version of the user guide committed on average 31.9 errors in comparison to the Experimental group who committed 10.9 errors. This represents a large divergence in the effectiveness of each group of users in performing tasks. Indeed, looking at individual error criteria (see Table 30) we can see some rather striking trends.

- Some 60% of Control group failed to complete one task (criterion 1)
- During the tasks, the Experimental group consulted the user guide 44.4% less than the Control group (criterion 2)
- The Experimental group remembered 84.2% more commands and icons than the Control group (criterion 3)
- The Experimental group made 69.2% fewer incorrect icon choices than the Control group (criterion 4)
- The Experimental group made 73.5% fewer incorrect menu choices than the Control group (criterion 5)
- The Experimental group did not ask any questions during the test whereas the Control group asked on average 1.8 questions (criterion 6)
- There were no observations of frustration in the Experimental group while in the Control group there were three individual observations of frustration (criterion 7)
- There were 76.9% fewer observations of confusion in the Experimental group (criterion 8)
- There was a single observation of satisfaction in the Experimental group compared with no observations in the Control group (criterion 9)
- The Control group used 58.8% more incorrect commands or input (criterion 10)
- In the Control group, two participants mistakenly thought they had completed one or more tasks; in the Experimental group, all participants successfully completed the tasks before finishing work (criterion 11)



Criterion	1	2	3	4	5	6	7	8	9	10	11
Control	0.6	3.6	9.3	2.6	6.8	1.8	0.6	2.6	0.0	3.4	0.6
Experimental	0.0	2.0	4.5	0.8	1.8	0.0	0.0	0.6	-0.2	1.4	0.0

Figure 22: Average Usability Criteria Error Scores for Each Group in Main Study

Looking at the total average error rates for each group (Figure 23), we can see that the Experimental group, using the user guide with Iconic Linkage, committed on average 10.9 errors throughout the test. In comparison, the Control group, which was using a user guide without Iconic Linkage, committed on average 31.9 errors over the course of the test. Putting this in perspective, we can say that the Experimental group made 65.8% fewer mistakes than the Control group.

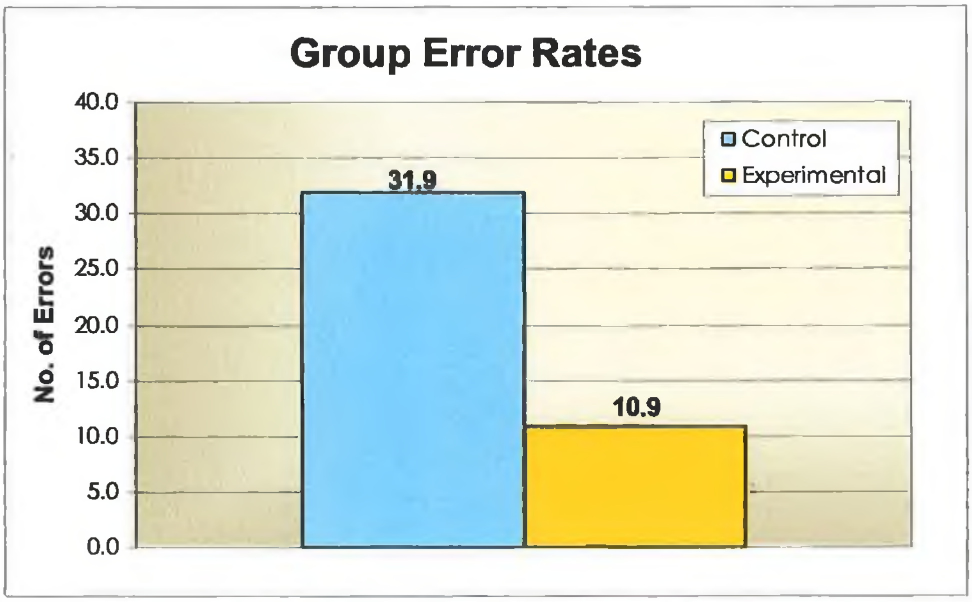


Figure 23: Average Group Error Rates in Main Study

5.5.4.3 *Results of QUIS Usability Survey*

A two-sided Wilcoxon-Mann-Whitney exact test of the null hypothesis of no difference between the average overall score for the Control group and Experimental group yielded a *p-value of 0.0079* indicating that there is a statistically significant difference between the average scores across the two groups. Examining the graph of average responses from the QUIS questionnaire (Figure 24), it immediately becomes apparent that there are a number of areas where there is no major difference visible between the two groups. The following paragraphs will examine these areas.

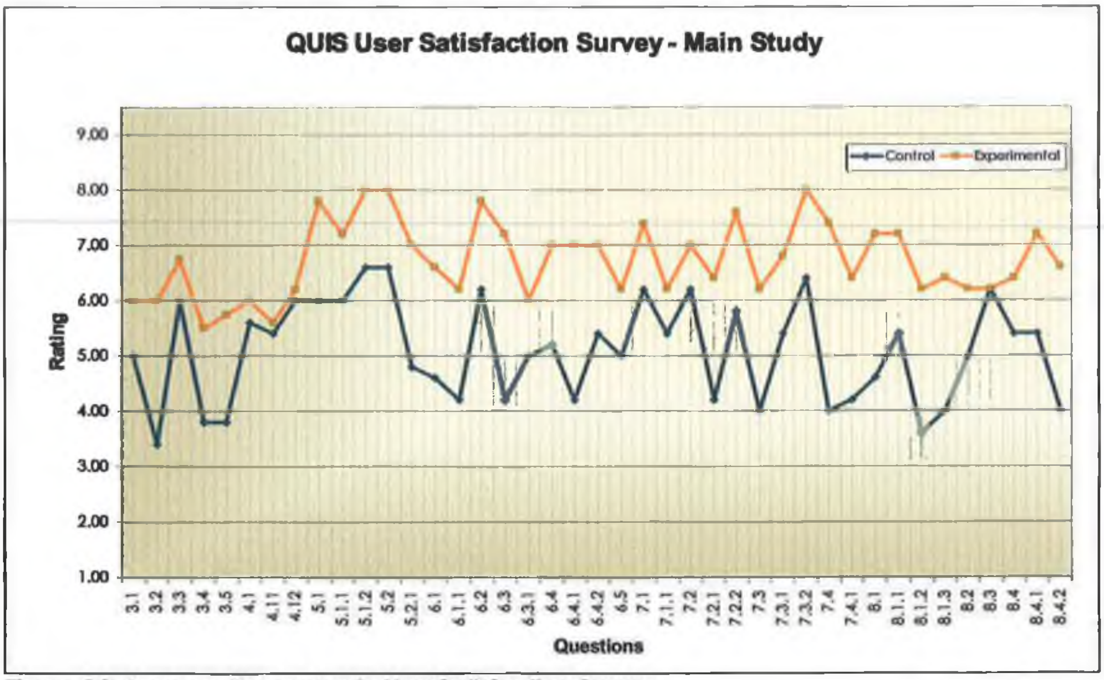


Figure 24: Average Responses in User Satisfaction Survey

Questions 4.1, 4.11 and 4.12

- Page layout was helpful: never/always
- Amount of information displayed on a page: inadequate/adequate
- Arrangement of information on page: illogical/logical

The results for these questions converge between the two groups. The fact that there is no noticeable difference between the two groups can be explained in that the questions relate to the amount of information on the page and the layout of information on the page. Neither of these factors were altered in the study. In fact, every effort was made to ensure that these factors were identical in both versions.

Question 8.3

■ Remembering names and use of commands: difficult/easy

This question relates to how easy subjects found it to remember the names and use of commands. The two groups rate this almost identically indicating that they both felt that they could remember this information with above average ease. However, it is apparent from the Post-Test Survey (PTS) scores for the two groups that this is most certainly not the case. With an average PTS score of 9.3, the Control group clearly found it more difficult to remember commands than the Experimental group with a PTS score of 4.5. To put this in context, Control subjects remembered on average just 5.7 commands whereas subjects in the Experimental group remembered on average 11.5 commands; the Experimental group performed roughly twice as well as the Control group.

In the other sections of the questionnaire, we can see quite distinct differences between the satisfaction levels of both groups with regard to the various aspects of the user guide, product and tasks. In the following paragraphs we will examine these sections.

Section 3: Overall User Reactions to System

terrible/wonderful: The Control group gave a rating of 5 on average. The Experimental group were 20% happier with average rating of 6.

frustrating/satisfying: The Control group found it on average frustrating, awarding it an average rating of 3.4. The Experimental group were approximately 76.4% more satisfied with an average rating of 6.

dull/stimulating: The Control group found it on average stimulating, awarding it an average rating of 6. Experimental group were 12.5% happier awarding it an average rating of 6.75.

difficult/easy: With an average rating of 5.5, the Experimental group found the system almost 44.7% easier to use than the Control group which gave an average rating of 3.8.

rigid/flexible: The Control group found the system more rigid than the Experimental group with average ratings of 3.8 for the Control group and 5.75 for the Experimental group.

While these criteria do not directly relate to the incidence of Iconic Linkage (IL), they clearly show that the increased usability caused by the inclusion of IL results in improved user attitudes towards a product. In this case it is clear that users have a better opinion of the software simply because the user guide is more usable.

Section 5: Terminology

This section deals with the terminology used in the user guides. On the whole, both groups felt that the level of terminology used in the user guides was appropriate and consistent. The Control group did, however, feel that computer terminology was used slightly too frequently. Bearing in mind that both user guides contained exactly the same information and that if anything, terms were more likely to be repeated in the Experimental version, it is difficult to understand how this perception arises. It could be presumed that the overall difficulties encountered by subjects resulted in "spill-over" into other areas of the user guide with the result that difficulties in one area can result in lower levels of satisfaction throughout the user guide in a form of "guilt by association" effect.

Section 6: User Guide

For *Question 6.1* "The user guide is: confusing/clear" the average Control group rating was 4.6 while the average Experimental group rating was 6.6. This indicates that the Experimental group found the user guide 43.5% clearer than the Control group did.

In response to *Question 6.1.1* "The terminology used in the user guide: confusing/clear", the average Control group response was 4.2 and the average Experimental group response was 6.2. Even though the terminology used in both versions was almost identical, these figures represent a 47.6% improvement in the Experimental group. One possible explanation for this is the effect on users' overall attitudes to the document caused by improved usability and accessibility of information.

Question 6.2 "Language used in the user guide is consistent: never/always" provided some interesting results in terms of how noticeable the repetition of textual structures was perceived by subjects. The average Control group response to this question was 6.2 and the average Experimental group response was 7.8. While on the one hand this shows that the Experimental version of the user guide was more consistent than the Control version, the fact that there was no large increase in consistency from the point of view of the user indicates that users were not particularly aware of the repetition of textual structures. As such, we can argue that Iconic Linkage is not as noticeable and therefore not as disruptive to users as one would imagine. On a related issue, however, one subject in the Control group mentioned in the post-task interview during the pilot study, without any form of prompting or specific questioning, that the lack of consistency and Iconic Linkage (the subject did not use this term but rather described the phenomenon) in the user guide proved problematic and distracting and hindered comprehension. Although no other subject mentioned this (the information was not solicited) it does point to the fact that the presence of Iconic Linkage is not always noticed by readers whereas a lack of Iconic Linkage is noticed.

In response to *Question 6.3* "Understanding information in the user guide is: difficult/easy" the Experimental group with an average rating of 7.2 found the information in the user guide 71.4% easier to understand in comparison to the Control group which responded with a rating of 4.2.

For *Question 6.3.1* "Finding a solution to a problem using the user guide: impossible/easy" the Experimental group with an average rating of 6, found it on average 20% easier to find solutions to problems than the Control group with a rating of 5.

In responding to *Question 6.4* "Amount of help given: inadequate/adequate" the Control group gave an average rating of 5.2 and the Experimental group gave a rating of 7. From this we can see that the Experimental group were 34.6% more satisfied with the information provided by the user guide even though both versions contained the exact same information. Again, we can see that improved usability in one area can improve overall subjective ratings in other areas.

In *Question 6.4.1* "User guide defines specific aspects of the system: inadequately/adequately" with an average rating of 7, the Experimental group found the user guide's definition of specific aspects of the system 66.6% better than the Control group who gave an average rating of 4.2.

In response to *Question 6.4.2* "Finding specific information using the user guide: difficult/easy", the Experimental group with a rating of 7 found it 29.6% easier to find specific information using the user guide than the Control group which gave an average rating of 5.4.

On the basis of the responses from the two groups to *Question 6.5* "Instructions for performing tasks are clear and unambiguous: never/always", the Experimental group (6.2) found the instructions 24% clearer and easier to understand than the Control group (5).

Section 7: User Guide Content & Structure

In response to *Question 7.2.1* "Information for specific aspects of the system was complete and informative", the Control group gave an average rating of 4.2 compared to the Experimental group which gave a rating of 6.4. While both user guides had identical information, the apparent difficulty encountered by subjects in finding information implies that they had problems assimilating or recognising information and as such could not recognise the information content of what they were reading.

For *Question 7.2.2* "Information was concise and to the point" the average Control group rating was 5.8 compared to the Control group's rating of 7.6. The above average ratings for the Control group can be interpreted as meaning that the presentation of information in the Control user guide (despite the slightly higher word count) was not substandard, i.e. it was of an acceptable standard. The Experimental user guide, however, was more than acceptable in comparison.

For *Question 7.3* "Tasks can be completed: with difficulty/easily" the average rating for the Control group was 4 while the Experimental groups average rating was 6.2. This represented a 55% improvement in the Experimental group's rating.

This result is borne out by task completion figures for both groups. In the Control group, three out of the five subjects (60%) failed to complete all of the tasks. In comparison, all of the subjects in the Experimental group completed all of the tasks.

For *Question 7.3.1* which asks whether subjects found the instructions for completing tasks clear or confusing, the Experimental group produced an average response of 6.8 in contrast to 5.4 for the Control group. The Experimental group, therefore found the instructions 25.9% clearer.

For *Question 7.4*, "Learning to operate the system using the user guide was: difficult/easy" the Experimental group gave an average rating of 7.4 compared to the average Control group rating of just 4. This represents quite a drastic difference with the Experimental group finding it 85% easier to learn to use the system.

Question 7.4.1 "Completing system tasks after using only the user guide was: difficult/easy" provided similar results with the Control group giving an average rating of 4.2 compared to 6.4 given by the Experimental group. Again, these figures are echoed in the error rates for each group and the task completion rates.

5.6 Conclusions on Empirical Study

In this chapter we have established that in order to test the hypothesis that Iconic Linkage improves the usability of software user guides, some form of summative evaluation involving users is essential. With our definition of usability consisting of both quantifiable and subjective components, the need to collect both quantitative and qualitative data is apparent.

This chapter discussed the various methods for collecting data and the ways in which they can be implemented. It is clear from this chapter that indirect observation is preferable over direct observation methods because of the risk of influencing subjects. However, the nature of the tasks and the facilities available for setting up a usability laboratory made it impossible to conduct the experiment without the administrator being present in the laboratory. While this is less than ideal, the effect of the administrator's presence was minimised through careful positioning and regulated interactions during the experiments.

We also considered data collection methods such as interviews, video and audio recording, screen logging and questionnaires. After discussing each of these in detail, it was possible to select and reject methods on the basis of their suitability for the requirements of this study.

An examination of literature relating to previous experiments and case studies was carried out in the hope of finding useful information on conducting usability experiments. What emerged from this review is that there seems to have been a shift away from documentation usability testing over the past decade or so, particularly with regard to print documentation. Those that do deal with documentation, regard documentation as including both print and online texts. Other studies which exhibit certain compatibilities with this study often differ in terms of their goals and objectives or they make inappropriate assumptions. Nevertheless, by analysing several studies, it was possible to extract useful pointers for conducting a usability experiment. Of the literature reviewed, only two stand out as being particularly relevant or useful. These studies were discussed in detail.

With this background knowledge, the chapter proceeded to describe the preparations, procedures and results of a pilot study conducted to test the methodology and protocols for the study. This consisted of producing materials and forms, recruiting participants, defining evaluation criteria for testing the user guide etc. The chapter describes the problems encountered during the pilot study. One such problem which emerged related to the specification of evaluation criteria. This proved problematic because although certain criteria may be useful or important, they may not necessarily be measurable due to the nature of the product and tasks. Similarly, data collection tools and methods are not always suitable for recording a particular type of information. Consequently, a number of changes had to be made before conducting the main study.

Other issues such as those encountered with participant P4 show that a great deal of preparation, flexibility and discipline on the part of the tester are essential in order to cope with unforeseen eventualities. Future research of this nature would need to take into account the notion of field-dependent and field-independent people, i.e. people who are more or less likely to use a user guide to learn how to use software. Screening of participants would, therefore, need to identify whether potential participants tend to use user guides or whether they prefer to "figure it out for themselves". This could be done simply by asking them how they normally learn how to use software or by using the *Group Embedded Figures Test* – GEFT (Witkin *et al.* 1971).

Having described the pilot study and the modifications made to the test procedures, the chapter proceeded to discuss the main study. From this we can see quite clearly that Iconic Linkage clearly has a positive effect on the usability of software user guides.

Taking the first of the three components of usability, i.e. the speed with which users perform tasks, the results of the empirical study clearly show that subjects using a user guide into which Iconic Linkage has been introduced performed tasks significantly faster than those using a user guide with no Iconic Linkage.

An interesting issue arises in relation to the results of the pilot study and the main study with regard to task times. While both studies showed the same dichotomy of results between the two groups, the subjects in both groups in the main study performed the tasks significantly more quickly than their counterparts in the pilot study. This can be explained by the fact that the participants in the main study were more experienced in the type of work involved, had more experience of using computers and had a better understanding of the context in which the software is used than those in the pilot study.

With regard to error rates for the two groups in the main study, the results show that the Control group made three times more mistakes than the Experimental group. Considering this more closely we can see that the Experimental group completed more tasks, worked more efficiently and made fewer mistakes using icons and commands. The Post Test Survey also shows that the Experimental group remembered more information about the software than the Control group.

In assessing the final component of our definition of usability, the user satisfaction questionnaire shows that attitudes to and satisfaction levels with the software were considerably more favourable in the Experimental group than in the Control group. The Experimental group found that the user guide which featured Iconic Linkage was clearer, easier to understand and more effective in helping them achieve their goals than the Control group. Interestingly, questions in the questionnaire designed to detect whether users detected the presence of Iconic Linkage indicated that the Experimental group did not detect Iconic Linkage. Both groups gave broadly similar ratings for the consistency and amount of repetition. This indicates that introducing repetition into a user guide does not necessarily represent a distraction for users. Indeed, one subject in the Control group actually commented that the lack of consistency in phrasing instructions was problematic for comprehension. No other subject mentioned this but this is a definite reference to Iconic Linkage – the user said that the lack of Iconic Linkage was distracting and resulted in the need to "refocus" after each sentence.

In the main study, another interesting issue arose in relation to interference between users' existing knowledge and the new information they were trying to learn. The existing knowledge domain in question related to users' prior knowledge of *Microsoft Word* and *QuicKey*. One member of the Control group noted that the way *QuicKeys* worked in a fundamentally different way to the way *Microsoft Word* implements a similar function. Thus, prior knowledge hampered users' learning of the new information. This information was not specifically requested, it was volunteered by the subject; no other subjects mentioned this.

Overall, the empirical study shows that Iconic Linkage is a viable and effective strategy for improving the usability of software user guides. Introducing Iconic Linkage into a text helps users understand the information more easily and learn to use the software more quickly. Even with the small sample sizes used in this study, it has been shown with a high level of statistical reliability that Iconic Linkage makes user guides more effective, that users can perform their tasks more quickly and retain more knowledge for longer.

The fact that clear improvements were detected across the three components of the test, i.e. task times, error rates and satisfaction levels, also shows that concerns regarding the possible confounding influence of the marginal improvement in readability or elimination of passive sentences are unfounded. As was discussed on page 198, if the absence of passive sentences represented the sole improvement in the user guide, the improvements would be restricted to the task times only because participants would have been able to read the user guide more quickly. However, because error rates and satisfaction levels also improved among the Experimental group, it is unlikely that this is simply due to the lack of passives. As such, it is difficult to treat the elimination of passives as a genuine confounding factor.

Chapter 6
CONCLUSIONS

6.1 Summary of Research

> Users' trust of systems is fragile; one experience with misleading data or unexpected results will undermine for a long time a person's willingness to use a system (Schneiderman 1998:13)

The stated aim of this study was to examine ways of improving the quality of software user guides. More specifically, the purpose of this study was to assess the effect of Iconic Linkage on the usability of software user guides. In the context of this study, user guides are regarded as a product of technical communication where technical communication includes both traditional technical writing and technical translation.

A key factor which is frequently overlooked by software designers, engineers and even users is that computers and user documentation are supposed to be tools which assist humans in doing something else, i.e. using a computer is not usually an end in itself. What is more, making sure that interface design complies with universal psychological facts is, according to Raskin (2000:4), "customarily omitted in the design process". This philosophy can also be applied to the production of user guides, which facilitate the use of these tools.

User documentation should reflect the needs, capabilities and limitations of the humans who use them. In order understand how to achieve this, the first task for this study was to take a detailed look at the genre of user guides. Chapter 2 introduced technical communication and examined the various facets of this sector before focussing on user guides. From this, it became clear that user guides are far more than just a repository containing every single piece of information relating to a product through which readers must trawl in order to find answers to their questions

(Weiss 1985:7). Rather, as Weiss puts it, they are similar to devices which must be designed, structured and built so that they perform a specific function for readers: to deliver precisely the kind of information users need, when they need it. User guides are structured, modular tools which facilitate users in their goal of learning how to use a piece of software.

With this focus on the needs of users, particularly with regard to information content, sequencing, presentation, delivery and availability, it is clear that traditional methods for assessing the quality of user guides are less than satisfactory. Indeed, Chapter 2 highlighted the fact that well known readability tests such as the Flesch scale reflect only a small part of the effectiveness of user guides. In fact, Klare, who was instrumental in the development of such tests, later admitted that such tests were never intended to be used on user guides and that he was unaware that they were even being used for such a purpose (Klare 2000:2-3). Chapter 2 concludes by providing us with a way forward: acknowledging that the success and ease with which readers (or users) can actually use a user guide should be the measure of quality, it leads us on to an investigation of usability.

Having defined usability 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" (see page 53), Chapter 3 began by discussing the most fundamental component of usability – humans. More specifically, it examined human cognition, a system which is pivotal to our ability to use a user guide. Here we explored the various systems and aspects that govern the way numerous marks on a sheet of paper are transformed into meaningful information which we need in order to learn how to use software. From our perceptive processes to cognitive mechanisms and learning capabilities, it is clear that human cognition is a complex information processing system which has many powerful capabilities but which has several important limitations.

Armed with this knowledge, Chapter 4 examined ways of harnessing the capabilities of human cognition and compensating for its limitations. In this chapter, we also looked at how we can identify those parts of an interface or interaction which can pose problems for humans.

In doing so, it was necessary to define user guides in terms appropriate to the area of cognetics – the process of engineering interfaces to match human cognitive abilities. As such, the notion of the user guide as an interface is presented with the user guide being described as an interface between the user and the software. Consequently, the same principles of interface design so often reserved for traditional software interfaces find applications in printed documents.

The chapter then introduced the notion of principles, guidelines and rules. Whereas principles draw on an understanding of human cognition to identify areas of concern, guidelines take this information and present goals or objectives which should be implemented to make interfaces as compatible as possible with human capabilities. Rules, then, provide definite strategies and methods for engineering the interface. Similarly, drawing on principles such as reducing STM load, taking advantage of recognition of information and humans' tendencies to form habits and chunk information, the concept of Iconic Linkage as a guideline is introduced.

Iconic Linkage was defined as the use of isomorphic constructions to express what is essentially the same information. Its origins in translation are discussed and comparisons are made with the concept of parallelisms used in technical communication. After providing examples of the various types of Iconic Linkage, i.e. partial, full, latent and introduced, strategies for implementing it are outlined. It is proposed that Iconic Linkage can improve usability in a number of ways including: aiding habit formation, reducing the need for problem-solving, improving the retention of information and reducing interference between tasks.

Having made the assertion that phrasing multiple instances of the same information using identical formulations improves usability, it is obvious that this needs to be tested empirically. On this basis, Chapter 5 discusses the different types of evaluation and identifies summative or diagnostic evaluation as the most appropriate approach for this study. After discussing numerous methods for collecting data, there follows an evaluation of models for conducting experiments based on previous studies. These models inform the design of an empirical study to test the effect of Iconic Linkage in a user guide, which is then reported on in detail.

6.1.1 Summary of Findings

From the analysis of data gathered during the empirical study it is clear that Iconic Linkage had a significant effect on the performance of the Experimental group. The Control group took nearly 80% longer to complete tasks and committed more than three times as many errors than Experimental group. The post test user satisfaction survey revealed that the Experimental group were almost a third happier with the user guide and product than the Control group. In addition, the Experimental group remembered twice as much information from the user guide.

These results show that the presence of Iconic Linkage in a user guide makes it easier for users to learn new software and that they will work more effectively and remember more information afterwards. This has ramifications for companies because the improved usability of user guides and the increased product satisfaction as a result may lead to greater customer satisfaction and ultimately loyalty.

6.2 Evaluation of Empirical Method

In testing the hypothesis that Iconic Linkage improves the usability of software user guides, a method was chosen which saw Iconic Linkage implemented in a situation which closely approximated real use and which involved human subjects.

A review of previous studies unearthed a number of surprising findings. The first of these was that there is a serious lack of previous empirical studies aimed specifically at assessing printed user guides. This apparent lack of experimental work can be attributed in part to the second finding of the literature review which indicates that there is a pervasive notion that print and online documentation can be assessed using the same procedures and criteria. In this regard, this study not only assesses the effect of Iconic Linkage, it also addresses a significant gap in research in relation to usability testing.

Of the studies reviewed, only two proved compatible with the aims of this study: Sullivan & Chapanis (1983) and Foss *et al.* (1981). Using these studies as a broad framework for the empirical study, a variety of tools, methods and procedures from other sources were incorporated to produce an evaluation model that draws on established best-practices from theory and research and incorporates into this, modern technology, tools and practical expertise. In this sense, the experimental model represents a new approach to the usability testing of printed documentation in that it combines academic research methods with industry-based practices.

This is also evident from the numbers of participants chosen to take part in the study. While traditional academic research involving subjects generally requires large numbers of subjects in order to provide significant results, industry-based evaluations use significantly fewer people, sometimes as a result of financial and time constraints.

Where such an approach would traditionally damage the credibility of the results, this study has shown that it is indeed possible to implement usability evaluations using low numbers of participants and still obtain statistically significant results.

This model is all the more reliable and effective at determining relative levels of usability in printed documentation because it assesses usability on three fundamental levels: the speed with which users work, the number of errors they make and the level of user satisfaction after performing tasks. Frequently, studies concentrate on one or two of these factors but to get a true picture of usability, all three need to be incorporated into a study. Indeed, Frøkjær *et al.* (2000) have shown this to be true particularly when we consider that there appears to be no correlation between the three: simply because users work faster, does not mean that they are making fewer mistakes and similarly, users who work more effectively with a product, do not necessarily prefer that product.

Another advantage of the method used in this study is that it is a repeatable and relatively low-cost method of user-based testing. While the study took place over three weeks, the total number of hours required with subjects was 18 hours. Even at a generous hourly rate of pay, the payments to subjects are not huge. Similarly, the cost for tools required to perform such a study are also quite low. The only tools needed are the QUIS questionnaire, Camtasia and a video camera.

Even where different aspects of a product are being assessed, the general structure of the evaluation model can still be used even though different tasks and criteria may be involved.

6.2.1 Limitations of Empirical Study

Although the strengths and benefits of the study and its findings as outlined above are clear, it can be argued that the model for implementing and evaluating Iconic Linkage used in this study is not without its limitations. Some of these limitations are limitations in the sense of possible weaknesses and are caused by certain circumstances and choices. Others, are not limitations in the strictest sense, but rather represent areas where the model could be refined or improved.

Numbers of Subjects

One of the most significant areas where the model could be refined is the number of subjects used in the empirical study. Although the aim of this study is not a formative, developmental investigation but rather a summative evaluation, some of the methods used are, nevertheless, of a formative nature. This was deemed necessary, firstly, because of the need to achieve a high degree of realism from and industry point of view, and secondly, because it would not have been possible to use large numbers of participants in the empirical study. Factors which played a role in this decision included cost considerations, the availability of suitable participants and time constraints affecting both the pilot study and the main study.

While using 5 subjects in each group may be acceptable from an industry point of view, an ideal situation would see significantly more participants used. Even though Wilcoxon-Mann-Whitney exact tests can be used to reliably determine the statistical significance of results obtained from small sample sizes, the use of larger numbers of subjects would allow the use of a wider range of statistical tests. Larger numbers would also allow us to analyse data in different ways, such as analysing the ratings for each individual question in the QUIS questionnaire rather than overall satisfaction ratings.

Nevertheless, the numbers used in this study still allow us to analyse the three sets of data and determine the statistical significance of the data obtained.

Screening of Subjects

Another area where the empirical method could be improved is in the screening process for selecting participants. The method chosen was based largely on the user profile produced in conjunction with the company that manufactured the test software. The first two parts of the QUIS questionnaire were also used as an indicator of how well participants backgrounds matched.

However, despite recruiting candidates who met all of the criteria and provided similar if not identical answers to the questions regarding PC experience, it was clear that there were certain differences among members of each group in terms of how they tackled tasks and responded to problems, as well as their general demeanour during the tests. It is uncertain whether these differences are due to certain differences in terms of PC experience, individual problem-solving strategies or indeed, personality traits. Nor is it certain whether these differences could have affected subjects' performance in the tests. However, because many of the differences appeared to manifest themselves quite equally between the two groups, it can be presumed that no one group was at an advantage and that neither group had "better" subjects.

It could also be argued that such heterogeneity of subjects might actually be a more accurate reflection of real users than completely homogeneous groups. This issue would certainly merit further experimental evaluation at a later stage.

Another issue relating to the screening of participants was apparent from the problems encountered with participant P4 in the pilot study. As mentioned on page 243, the study did not take into account the fact that participants may be defined as *field-dependent* or *field-independent* and this can have implications for the way in which they use user guides. Consequently, the experimental methodology used here should be supplemented with some form of screening to identify whether potential participants tend to use user guides or whether they prefer to adopt a more independent, problem-solving approach. This could be done by asking them how they normally learn how to use software or by using the *Group Embedded Figures Test – GEFT* (Witkin *et al.* 1971).

Subjectivity of Certain Error Criteria

Certain error criteria used in the study are inherently subjective and are open to wide interpretation. Such criteria include observations of frustration, confusion and satisfaction. Without some form of think-aloud protocol, it is extremely difficult to assess what a user is actually thinking at a particular moment. As such, attempting to guess a users emotional state by simple observation is problematic. While in this study, these observations were registered if there was a clear change in body position accompanied by a facial expression or verbalisation, this is far from ideal. Future studies would need to make some provision for this subjectivity and perhaps, involve two or more people observing the videotapes; only if all observers agreed, would an observation be counted.

Selecting Tasks

The nature of the tasks used in the empirical study is an area worthy of further investigation. Although the tasks are largely dependent on the software on which they are based, it would be useful to be able to select and structure tasks in such a way that we can see which aspects of the tasks are most affected by Iconic Linkage. If this proved possible, the empirical model would then be able to tell us which tasks and types of tasks would benefit the most from Iconic Linkage. This would give us a better understanding of which cognitive functions are best served by Iconic Linkage and it would also make it easier to implement Iconic Linkage in a more focussed way.

Implementing Iconic Linkage

In this study, our method for implementing Iconic Linkage involved the use of style guides and Computer-Aided Translation (CAT) tools. This can present some questions as regards the effectiveness and efficiency of the process and the quality of the results.

With regard to style guides, much depends on the quality and accuracy of the guidelines they contain. If a style guide consists largely of poor recommendations or ungrammatical rules, we may be left wondering what effect this will have on a user guide which has been edited on this basis. Is it possible for a user guide to be usable as a result of implementing Iconic Linkage with a style guide and still be poorly written? Ultimately, we can assume that the quality of the Iconic Linkage in a user guide depends on the quality of the rules governing its implementation, i.e. the style guide. It would be interesting, however, to determine the effect, if any, on the usability of a user guide as a result of using a substandard style guide.

CAT tools were also used to implement Iconic Linkage in this study. While this is highly effective, it poses a number of practical problems which relate to their use in industry. CAT tools are generally the preserve of translators who use them on completed texts. And because CAT tools are not authoring tools but rather tools for processing existing texts, it is unclear how technical authors producing texts will be able to use them, even though the majority of CAT tools are able to handle a wide range of file formats. It is more likely, that in practice, CAT tools would have to be used as part of a subsequent editing, revision or translation process rather than during the initial writing phase.

6.3 Implications & Applications of this Study

In discussing the implications and possible applications of this study and its findings, it is useful to remind ourselves of the wider context in which this study was conducted. Our analysis of user guides firmly places the study in the field of technical communication. However, on a practical and industrial level as well as on a theoretical level, it is important that we define technical communication here as including technical translation. This is due in part to the virtual inevitability of translation for user guides. It is also due to the fact that technical translation aims to produce technical documents for a new target audience, albeit in a different language. As such, technical translation is simply another environment for the production of user guides. It is thus, equally justified in being included here.

With this backdrop to the study, we can categorise the implications of this study on the basis of the two basic components that make it up: the hypothesis and the empirical study. Our hypothesis is that Iconic Linkage improves the usability of software user guides while our empirical study includes the various methods, criteria and procedures developed to test the hypothesis.

Both components have implications and possible applications for a range of areas but we can group them under the following categories:

- Implications for Practice
- Implications for Research
- Implications for Teaching

6.3.1 Implications for Practice

Perhaps the most obvious implication for this study is that it provides a sound theoretical basis as well as empirical data to prove the effectiveness of style conventions, guides and recommendations. Rather than using such conventions, guides and recommendations because they are "common practice" or perceived as "best practice" without any real justification for doing so other than simply making texts clearer and more readable, this shows that they should be used because they make texts more usable. Indeed, as we saw in Chapters 4 and 6, usability is a much more comprehensive and ultimately more valuable goal than readability. Instead of being a collection of preferred methods of phrasing information, style guides are, in effect, cognitive engineering tools.

As a guideline itself, Iconic Linkage represents a single strategy which, when implemented in a user guide or its translation, can produce significant improvements in the usability of documents. It should be pointed out here that while this study was concerned with software user guides, there is no reason why Iconic Linkage could not be used to produce the same benefits in any kind of instructional text dealing with any subject. The principles of technical communication, human cognition and learning processes remain the same regardless of the subject or document type. This also applies to the practice of single-sourcing. As previously mentioned, this practice sees documentation being produced from a central "stock" of text to produce user guides, help files, web pages, marketing materials and so on. It is clear that the implementation of IL will have certain implications for such a practice. The most obvious one is that the consistency and clarity of texts containing IL would transfer over into the other types of documentation created from the master text stock. However, because each of these types of documents has its own particular requirements and conventions, further research in the form of usability studies would be essential in order to determine the impact of IL on the quality and usability of single-source documentation.

Iconic Linkage also has an impact on the translation of user guides. Since Iconic Linkage essentially involves introducing consistency and repetition into a text and limiting the ways in which information can be phrased, it has a significant effect on the effectiveness of translation memory tools. These tools are essentially databases which are used to store sentences and their translations. The idea is that as each sentence is translated, it is stored along with the translation so that if the same sentence is encountered again during translation, the existing translation is retrieved from the database and inserted into the text, thus negating the need to translate the sentence a second time. Obviously, if the same sentence or part of it is repeated several times in a text, it will only have to be translated once.

The benefits of this to industry include faster turnaround times for translations, greater reusability and consistency of translations. And because repetitions reduce the total number of words to be translated, the overall translation costs are reduced. Add to this the improved usability introduced by Iconic Linkage and we can see how beneficial Iconic Linkage is to the translation industry. Indeed, it could be argued that the practical benefits of Iconic Linkage for the translation of user guides are greater than they are for the monolingual production of user guides. Ultimately, however, many of the benefits feed back to the software manufacturer in the form of shorter times to market, reduced costs and improved products.

But it is not just our findings that benefit technical communication practice. The empirical model with its procedures and criteria offers practical benefits for the production and translation of user guides.

The list of criteria used to evaluate the usability of user guides (see Chapter 6) can be used to plan, design and assess user guides both in traditional technical communication contexts and in translation contexts. Furthermore, the test procedure also represents a reusable and relatively low cost model for assessing the quality and usability of documents as well as for highlighting other issues in user guides.

To summarise, Iconic Linkage is a relatively simple guideline which can be implemented using a variety of style rules by technical writers, translators and editors to produce more usable user guides. User guides which feature Iconic Linkage are easier to read, understand and use and they allow users to work more quickly and effectively and remember more information for longer.

The empirical method allows anyone involved in the production and translation of user guides to assess the documents on the basis of user expectations and requirements easily and with a minimum of expense.

6.3.2 Implications for Research

One of the most significant implications of this study from a theoretical perspective is that it provides new impetus for research into writing methods and strategies in technical communication. For many years, the emphasis of much research in technical communication has been on the usability of delivery methods and the type of information presented in documentation and not on the usability of the information itself. With this study, the focus is placed back on the way information is formulated. The study challenges the prevailing trend in technical communication research which seeks to treat print and online documentation as one and the same.

We also saw in Chapter 6 that testing the usability of documentation, particularly print documentation, has become much less fashionable as software developers with technical communicators in tow concentrate on flexibility in the misguided belief that it has superseded usability. Indeed, it can be argued that there is a prevailing belief in some quarters that flexibility has solved the problem of usability and rendered it obsolete (Weiss 1995:13; Mehlenbacher 1993:210).

On a related note, the study clearly shows that readability is not a reliable measure of the quality of documents. The two versions of the user guide used in the empirical study both achieved very similar scores in readability tests. This similarity does not, however, reflect the differences in user performance as a result of using these user guides. This study, therefore, indicates that there is no link between readability and the success of user guides. A user guide that is readable is not necessarily a "good" user guide.

This study shows that textual factors, as opposed to technical factors relating to the delivery of documentation, can produce improvements in usability worthy of note and which could be utilised in many other areas of technical communication and translation.

But it is not just in the area of technical communication where the findings of this study can have an impact. Research into translation theories and practices can also draw benefits from this study on a number of levels.

Iconic Linkage and the study of usability provide new insights and new ideas for the way we look at theories of translation. While these areas are of practical importance to translators in the production and assessment of better quality texts they also support paradigms for discussing translation which do not fall within the scope of equivalence or Skopos theories. Rather, it builds on the notion of Skopos theory as proposed by Vermeer (1978) and indeed aspects of relevance theory (Sperber & Wilson 1986; Gutt 1991) and provides the basis for a user-centred approach to examining translation. With such an approach, the idea of translating non-isomorphic but semantically identical sentences with the same text challenges certain theories of translation which state that the nuances and "flavour" of the original must be preserved. Indeed, rather than being "concerned with the source text and with its inviolable 'sanctity'" whereby target text factors "while never totally ignored, often counted as subsidiary" (Toury 1995:24) this approach unashamedly concentrates on the target audience's needs and how these needs are catered for in the translated text.

The concept of Iconic Linkage also highlights new areas for research into CAT tools. We saw how one such tool, Trados Translators' Workbench, was used to implement Iconic Linkage in one version of the user guide. However, this process was not without its difficulties, chiefly because of the tool's inability to identify semantically identical sentences. While CAT can identify Iconic Linkage in a text, and indeed benefit immensely from it (CAT is designed to work best when there are identical sentences in a text), an ideal situation would see CAT play more of a role in introducing Iconic Linkage into a text in the first place. To do this, the technology used in CAT tools would need to be re-evaluated to see how CAT tools can "understand" text in order to identify semantically identical information.

Another area which is closely linked to the notion of Iconic Linkage is that of Controlled Language (CL) which can be defined as follows:

> a subset of a natural language that is specifically designed for writing clear technical documentation in a particular domain [...] the subset is defined partly through restricted vocabulary, and partly through rules of composition (Power *et al.* 2003:115).

Thus, CL essentially is a set of rules governing the way in which information can be written. This concept is not unlike the notion of Iconic Linkage in that it restricts the ways in which information can be expressed. Indeed, Controlled Language generally uses rules not unlike those used in Chapter 6 to ensure semantically identical sentences were written in precisely the same way.

It is widely acknowledged that texts written in a CL are more readable and ultimately produce better results when used with a machine translation (MT) system (Møller 2003:95, 101) although these two aims are generally regarded as distinct goals of CL (Power *et al.* 2003:115). Nevertheless, we can see from this study that we can achieve significant improvements in document quality while at the same time making texts more suitable for use with translation tools. Admittedly, MT is beyond the remit of this study, but it is clear that like texts written in CL, texts that feature Iconic Linkage have the potential to yield better MT results than texts without Iconic Linkage. What is certain, however, is that such texts will be considerably easier to translate using CAT tools.

But what makes this study of particular relevance is that it supports the notion that CL improves the quality of texts from the point of view of readers. According to Møller (2003:95) "few reports on… usability tests of controlled languages have been published". Indeed, this is unsurprising considering the apparent concentration in CL circles on readability rather than usability. We are already aware of the unsuitability of readability as a measure of document quality.

6.3.3 Implications for Teaching

Although the implications of this study with regard to practical applications and research are by far the most significant ones, it is also worth considering the possible implications and applications of these findings for training purposes.

Technical Communication

In the field of technical communication teaching, the results of the empirical study provide concrete and tangible proof of the benefits of guidelines beyond the ethereal claims of improved readability. This can help students contextualise their learning and see a real application for their newly acquired skills. By showing students that good writing skills as typified by Iconic Linkage mean more than just a concise or clearly written text, but rather an effective and usable text, they can better appreciate the importance of writing skills and their role as technical communicators.

Similarly, the cognitive psychology background of this study provides a framework to help students understand the unseen mental processes involved in technical communication. Such knowledge helps to explain how human capabilities and limitations can be accommodated by implementing a single textual guideline.

Translation

From the point of view of training translators, this study provides clear evidence of the need for translators to understand technical writing, usability and cognitive psychology. This is particularly useful for technical translators who are a specific type of technical communicator. By better understanding the processes and principles that are involved in producing good technical documentation, translators will be able to ensure the quality of documentation through the translation process. And by introducing translation students to cognitive psychology and human factors and subsequently illustrating how this knowledge can be harnessed to make written communication more effective, students will gain enormous insight into their role as mediators or "explainers" of technical information.

BIBLIOGRAPHY

Anderson, J. R. (2000) *Cognitive Psychology and its Implications.* 5th edition. New York, USA: Worth Publishers

Apple Computer Inc. (1978) *Apple II BASIC Programming Manual, Informative and Entertaining Reading for Novice and Professional Alike.* Press release dated 8th June, 1978 [online] Available from: http://www-sul.stanford.edu/mac/primary/docs/pr4.html [Accessed: 16/04/02]

Asch, S. E. (1956) Studies of Independence and Conformity: A Minority of One against a Unanimous Majority. In: *Psychological Monographs,* 70, pp. 1-70

Austin, M. and Dodd, R. F. (1985) *The ISTC Handbook of Technical Writing and Publication Techniques: A Practical Guide for Managers, Engineers, Scientists and Technical Publications Staff.* London, UK: William Heinemann Ltd.

Bannon, L. (1991) From Human Factors to Human Actors The Role of Psychology and Human-Computer Interaction Studies in Systems Design. In: J. Greenbaum and M. Kyng (eds) *Design at work: Cooperative Design of Computer Systems.* Hillsdale, USA: Lawrence Erlbaum Associates, pp. 25-44.

Barrett, E. [ed.] (1989) *The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information.* Massachusetts, USA: The MIT Press

Beer, D. F. (1992) What Should You Write: A User's Guide, Tutorial, Reference Manual, or Standard Operating Procedure? In: *Manuals and Instructions: Giving Instructions that Work.* New York, USA: Institute of Electrical and Electronics Engineers [IEEE]

Boehm, B. (1996) A Spiral Model of Software Development and Enhancement. In: *ACM SIGSOFT Software Engineering Notes.* August 1986.

Borowick, J. N. (1996) *Technical Communication and its Applications.* New Jersey, USA: Prentice-Hall

Brainviews (2002) *The Animated Brain – Pavlov's Dog*' [online] Available from: http://www.brainviews.com/abFiles/AniPavlov.htm. [Accessed 12/10/02]

Broadbent, D. (1958) *Perception and Communication.* London, UK: Pergammon Press

Brockman, R. J. (1990) *Writing Better Computer User Documentation: From Paper to Hypertext – Version 2.0.* New York, USA: John Wiley & Sons

Buchanan, R. A. (1992) Textbase Technology: Writing with Reusable Text. In: P. O' Brian Holt and N. Williams (eds) *Computers and Writing: State of the Art.* Oxford: Intellect Books

Budin, G. (2000) *Terminology – Translation – Technical Communication: Lecture Notes.* [online] Available from: http://mailbox.univie.ac.at/~buding8/dint.htm [Accessed: 21/03/04]

Byrne, J. (1999) International English: Making A Case for Efficient and Clear Communication. In: *ATA Chronicle*, Vol. XXVII, No. 10

Byrne, J. (2003a) Translation, Quality and Cognetics: New Insights into Translation Quality. In: S. Cunico and B. Rodríguez de Céspedes (eds) *Proceedings Quality Issues in Translation.* University of Portsmouth. pp. 19-36

Byrne, J. (2003b) Evaluating the Usability of Software User Guides: A Pilot Study. In: Elisabeth Gräfe (ed) *tekom Jahrestagung 2003: Zusammenfassung der Referate.* Stuttgart: tekom GmbH. pp. 35-39

Calvin, W. H. (1996) *How Brains Think: Evolving Intelligence, Then & Now.* London, UK: Phoenix

Card, S. K., Moran, T. P. and Newell, A. (1983) *The Psychology of Human-Computer Interaction.* New Jersey, USA: Lawrence Erlbaum Associates

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. In: *Journal of the Acoustical Society of America*, 25, pp.975-979.

Coe, M. (1996) *Human Factors for Technical Communicators.* New York, USA: John Wiley & Sons

Cognitas (2003a) *Gute Bedienungsanleitung – geringeres Haftungsrisiko.* cognitas - Gesellschaft für Technik-Dokumentation: Paderborn [online] Available from: http://www.tuev-sued.de/presse/pressearchiv/iiftqseozlfy.asp. [Accessed: 21/03/04]

Cognitas (2003b) *DOCcert-Kooperationsvertrag bietet deutschen Herstellern erhöhten Schutz vor Bedienungsanleitungs-Reklamationen.* cognitas - Gesellschaft für Technik-Dokumentation: Paderborn [online] Available from: http://www.tuev-sued.de/start/pressearchiv/jtusowqgxxza.asp. [Accessed: 21/03/04]

ComputerBild (1999) Holen Sie sich beim Händler Geld zurück. In: *ComputerBild* 25/99, pp.12-17

Council of the European Union (1998) *Council Resolution of 17 December 1998 on operating instructions for technical consumer goods.* Luxembourg: Official Journal of the European Communities (98/C 411/01)

Curwen, C. (2002) Comments on "Give them Printed Documentation, too!!!". [online] Available from: http://www.tc-forum.org/topicus/ru31comm.htm. [Accessed: 22/05/2004]

D'Agenais, J. & Carruthers J. (1985) *Creating Effective Manuals.* Cincinnati, USA: South Western Publishing Co.

Davis, J. (1992) *How To Write a Training Manual.* Hants, UK and Vermont, USA: Gower Publishing Company

Deutsch, M. & Gerard H. B. (1955) A Study of Normative and Informative Social Influences on Individual Judgements. In: *Journal of Abnormal and Social Psychology*, 51, pp. 629-636

Dillon, A. (1992) Reading from paper versus screens: A critical review of the empirical literature. Ergonomics, 35(10), pp.1297-1326.

Dix, A. (1998) *Human-computer Interaction.* 2nd edition. New Jersey, USA: Prentice Hall

Downton, A. (1991) *Engineering the human-computer interface*, London; New York, USA: McGraw-Hill Book Co.

Dumas, J. S. and Redish, J. C. (1999) *A Practical Guide to Usability Testing.* Exeter, UK: Intellect Books

Ehn, P. & Löwgren, J. (1997) Design for Quality-in-use: Human-Computer Interaction Meets Information Systems Development. In: M. Helander, T. K. Landauer, P. Prabhu (eds) *Handbook of Human-Computer Interaction* (2nd edition). The Netherlands: Elsevier Science. pp. 299-313

Ellis, H. C. & Hunt, R. R. (1993) *Fundamentals of Cognitive Psychology.* 5th edition. Dubuque, Iowa: WCB Brown & Benchmark Publishers

EN 292-2: 1991 *Safety of machinery. Basic concepts, general principles for design.*

EN 62079: 2001 *Preparation of instructions - Structuring, content and presentation*

Erikson, E. (1965) *Childhood & Society.* Revised edition. Harmondsworth, UK: Penguin & Hogarth Press

Fabrizio, R., Kaplan, L., & Teal, G. (1967) Readability as a function of the straightness of right-handed margins. In: *Journal of Typographic Research*, 1, pp. 90-95

Faulkner, C. (1998) *The Essence of Human-Computer Interaction.* London, UK and New York, USA: Prentice Hall

Fitts, P.M. and Posner, M.I. (1967) *Human Performance.* Belmont, CA, USA: Brooks Cole

Foss, J. D., Rosson, M. B. and Smith, P. L. (1981) Reducing Manual Labor: An Experimental Analysis of Learning Aids for a Text Editor. In: *Proc. Human Factors in Computer Systems.* Washington: ACM. pp. 332-336

Fredholm, L. (2001) Article on Ivan Petrovich Pavlov. [online] Available from: http://www.nobel.se/medicine/educational/pavlov/readmore.html. [Accessed: 24/10/02]

Frøkjær, E., Hertzum, M. & Hornbæk, K. (2000) Measuring Usability: Are Effectiveness, Efficiency, and Satisfaction Really Correlated? In: *ACM Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, USA: ACM Press. pp. 345-352

Gavin, H. (1998) *The Essence of Cognitive Psychology.* Hertfordshire, UK: Prentice Hall Europe

Gerzymisch-Arbogast, H. (1993) Contrastive Scientific and Technical Register as a Translation Problem in Scientific & Technical Translation. In: S. E. Wright (ed) *Scientific and Technical* Translation: *American Translators' Association Scholarly Monograph Series*, Vol. VI 1993. Amsterdam, The Netherlands and Philadelphia, USA: John Benjamins

Ginsburg, H. P. & Opper, S. (1988) *Piaget's Theory of Intellectual Development.* 3rd edition. New Jersey, USA: Prentice Hall

Göpferich, S. (1993) Die translatorische Behandlung von Textsortenkonventionen in technischen Texten. In: *Lebende Sprachen.* No. 2/93, pp. 49-52

Gregory, M. and Poulton, E. E. (1970) Even versus uneven right-margins and the rate of comprehension in reading. In: *Ergonomics*, 13, pp. 427-434

Gutt, E. A. (1991) *Translation and Relevance: Cognition and Context.* Oxford, UK: Basil Blackwell

Harrison, S. & Mancey, P. (1998) Optimal amount of time for obtaining accurate usability-test results. In: *ACM Proceedings of the 16th annual international conference on Computer documentation.* New York, USA: ACM Press. pp. 173-179

Hartley, J., and Burnhill, P. (1971) Experiments with unjustified text. In: *Visible Language*, 5, pp. 265-278

Hartley, J. (1985) *Designing Instructional Text.* London, UK: Kogan Page

Heino, H. (1992) Spezialprobleme beim Übersetzen fachsprachliche Texte. In: *Fachsprache International Journal of LSP*, Vol. 3-4, pp.111-120

Helander M., Landauer T.K. and Prabhu, P. (eds) (1997) *Handbook of Human-Computer Interaction.* 2nd edition. Amsterdam, The Netherlands: Elsevier Science

Helsel, D. R. & Hirsch, R. M. (1992) *Statistical Methods in Water Resources.* Amsterdam, The Netherlands: Elsevier Publishers

Hill, S. (1995) *A practical introduction to the human-computer interface.* London, UK: DP Publications

Hoc, J. M., Cacciabue, P. C. and Hollnagel E. (eds) (1995) *Expertise and technology: cognition & human-computer cooperation.* Hillsdale, USA: Lawrence Erlbaum Associates

Houghton-Alico, D. (1985) *Creating Computer Software User Guides: From Manuals to Menus.* New York, USA and London, UK: McGraw-Hill

House, J. (1981) *A Model for Translation Quality Assessment.* 2nd edition. Tübingen: Gunter Narr Verlag

ISO 9241-11: 1998 *Ergonomic requirements for office work with visual display terminals – Part 11: Guidance on Usability*

IsoNorm Software Usability Questionnaire [online] Available from: http://www.sozialnetz-hessen.de/aweb/ph.asp?id=het [Accessed: 07/07/03]

Jacobsen, N., Hertzum, Morten & John, B. E. (1998) The evaluator effect in usability tests. In: *CHI 98 conference summary on Human factors in computing systems.* New York, USA: ACM Press. pp. 255-256

Johnson, P. & Johnson, H. (1991) Knowledge Analysis of Tasks: Analysis and Specification for Human-Computer Systems. In: A. Downton (ed) *Engineering the human-computer interface.* London, UK/New York, USA: McGraw-Hill. pp. 119-143

Jung, M. & Becker, O. (2003) *Die Bedienungsanleitung - kundenfreundlich und rechtssicher.* cognitas - Gesellschaft für Technik-Dokumentation: Paderborn [online] Available from: http://www.cognitas.de/pdf/tuev_cognitas.pdf. [Accessed: 21/03/04]

Kantner, L. & Rosenbaum, S. (1998) Current Issues in Assessing and Improving Documentation Usability. In: *ACM CHI 98 Conference Summary on Human Factors in Computing Systems.* New York, USA: ACM Press, pp. 389

Keirnan, T., Anschuetz, L. & Rosenbaum, S. (2002) Combining Usability Research with Documentation Development for Improved User Support. In: *ACM Special Interest Group for Design of Communications archive Proceedings of the 20th annual international conference on Computer documentation.* New York, USA: ACM Press, pp. 84-89

Kellogg, R. T. (1988) Attentional overload and writing performance: effects of rough draft and outline strategies In *Journal of Experimental Psychology: Learning, Memory, and Cognition.* Vol. 14, No. 2, pp. 355-6

Kieras, D. & Poulson, P. (1985) An Approach to the Formal Analysis of User Complexity. In: *International Journal of Man-Machine Studies, 22,* pp. 365-394

Kirakowski, J. (1990) *Effective methodology for the study of HCI.* Amsterdam; New York, USA: North-Holland

Kirakowski, J. (2001) *Usability Evaluation by Questionnaire.* London, UK: Taylor & Francis Group

Kirakowski, J. (unknown) *Questionnaires in Usability Engineering: A List of Frequently Asked Questions.* 3rd edition. [online] Available from: http://www.ucc.ie/hfrg/resources/qfaf1.html [Accessed: 28/08/02]

Kiraly, D. (2000) *A social constructivist approach to translator education: empowerment from theory to practice.* Manchester, UK: St. Jerome Publishers

Klare, G. R. (2000) Readable Computer Documentation. In: *ACM Journal of Computer Documentation,* Vol. 24, Issue 3, pp. 148-168

Koffka, K. (1935) *Principles of Gestalt Psychology.* London, UK: Routledge

Köhler, W. (1947) *Gestalt Psychology: An Introduction to New Concepts in Modern Psychology.* New York, USA: Liveright

Kreitzberg, C. (1998) *Ending the Software Struggle.* [online] Available from: http://www.cognetics.com/presentations/charlie/p_struggle.html [Accessed: 05/07/02]

Kreitzberg, Charles (1998) *The LUCID Framework – Overview.* [online] Available from: http://www.cognetics.com/lucid/lucid.html. [Accessed 19/02/03]

Landauer, T. K. (1995) *The Trouble with Computers: Usefulness, Usability and Productivity.* Massachusetts, USA/London, UK: The MIT Press

Landauer, T. K. (1997) Behavioral Research Methods in Human-Computer Interaction. In: M. Helander, T. K. Landauer, P. Prabhu (eds) *Handbook of Human-Computer Interaction.* 2nd Edition. Amsterdam, The Netherlands: Elsevier Science, pp. 203-227

Major, J. H. (1985) What Should You Write: A User's Guide, Tutorial, Reference Manual, or Standard Operating Procedure?. In: D. F. Beer (ed) *Manuals and Instructions: Giving Directions that Work*, IEEE. pp 121-124

Mamone, S. (2000) Documentation Testing. In: *ACM SIGSOFT Software Engineering Notes.* Vol. 25, No. 2 March 2000, pp. 26-30

Mancuso, J. C. (1990) *Mastering Technical Writing*, Menlo Park, California, USA: Addison-Wesley Publishing Company

Mehlenbacher, B. (1993) Software usability: choosing appropriate methods for evaluating online systems and documentation. In: *ACM Proceedings of the 11th annual international conference on Systems documentation.* New York, USA: ACM Press, pp. 209-222

Microsoft Press (1998) *The Microsoft Manual of Style for Technical Publications - Version 3.0.* [CD-ROM] Redmond, Washington: Microsoft Press

Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. In: *Psychological Review*, 63, pp. 81-97

Mirel, B. (1990) Usability and hardcopy manuals: evaluating research designs and methods. In: *ACM SIGDOC Asterisk Journal of Computer Documentation , Proceedings of the 8th Annual International Conference on Systems Documentation*, Volume 14, Issue 4, pp. 69-77

Molich, R. & Nielsen, J. (1990) Heuristic evaluation of user interfaces. In *Proceedings ACM CHI'90 Conference on Human Factors and Computing Systems.* Seattle, USA: ACM Press, pp. 249-256

Molich, R. & Wilson, C. (2003) New Tips and Tricks for a Better Usability Test. In: *ACM CHI '03 extended abstracts on Human factors in computer systems.* New York, USA: ACM Press, pp. 878-879

Møller, M. (2003) Grammatical Metaphor, Controlled Language and Machine Translation. In: *Proceedings of EAMT-CLAW 03: Controlled Language Translation.* Dublin, 15th-17th May, 2003. Geneva: EAMT, pp. 95-104

NECTAR (1999) *RESPECT User-Centred Requirements Handbook 5.3 – Part C: 4. User Requirements Methods.* NECTAR. [online] Available from: http://www.ejeisa.com/nectar/respect/5.3/415.htm [Accessed: 24/02/03]

Newell A. & Simon, H. (1972) *Human Problem Solving.* Englewood Cliffs, New Jersey, USA: Prentice-Hall

Nickerson, R. & Landauer, T. K. (1997) Human-Computer Interaction: Background Issues. In: M. Helander, T. K. Landauer, P. Prabhu (eds) *Handbook of Human-Computer Interaction.* 2nd edition. The Netherlands: Elsevier Science. pp. 3-31

Nielsen, J. & Levy, J. (1994) Measuring Usability: Preference vs. Performance. In: *Communications of the ACM.* Vol. 37, No. 4, pp. 66-76

Nielsen, J. (1994) *Usability Engineering*. London, UK and New York, USA: Academic Press

Nielsen, J. (2001) *First Rule of Usability? Don't Listen to Users*. [online] Available from: http://www.useit.com/alertbox/20010805.html. [Accessed: 29/06/02]

Norman, D. A. (1981) *Perspectives on Cognitive Science*. New York, USA: ABLEX Publishing Corporation

Norman, K. (2003) *Quant QUIS™: Information About Quantitative Analysis*. [online] Available from: http://www.lap.umd.edu/QUIS/QuantQUIS.htm. [Accessed: 19/09/2003]

Notess, M. & Swan, M. (2003) Predicting User Satisfaction from Subject Satisfaction. In: *ACM CHI '03 Extended Abstracts on Human Factors in Computer Systems*. New York, USA: ACM Press, pp. 738-739

Novak, J. D. (1998) *Learning, Creating, and Using Knowledge: Concept Maps™ as Facilitative Tools in Schools and Corporations*. New Jersey, USA: Lawrence Erlbaum Associates

Novick, D. G. (2000) Testing documentation with "low-tech" simulation. In: *Proceedings of the 18th Annual ACM International Conference on Computer Documentation: Technology & Teamwork*. Piscataway, New Jersey, USA: IEEE Educational Activities Department, pp. 55-68

O'Brien, S. (2003) Controlling Controlled English: An Analysis of Several Controlled Language Rule Sets. In: *Proceedings of EAMT-CLAW 03: Controlled Language Translation*. Dublin, 15th-17th May, 2003. Geneva: EAMT, pp. 105-114

Piaget, J. & Inhelder, B. (1969) *The Psychology of the Child*. London, UK: Routledge & Kegan Paul

Power, R., Scott, D. & Hartley, A. (2003) Multilingual Generation of Controlled Languages. In: *Proceedings of EAMT-CLAW 03: Controlled Language Translation*. Dublin, 15th-17th May, 2003. Geneva: EAMT, pp. 115-123

Preece, J. (1993) *A Guide to usability: human factors in computing*, Wokingham, UK and Massachusetts, USA: Addison-Wesley

Preece, J. (1994) *Human-computer interaction*. Wokingham, UK and Massachusetts, USA: Addison-Wesley

Prescott, J. & Crichton, M. (1999) Usability Testing: A Quick, Cheap, and Effective Method. In: *Proceedings of the 27th annual ACM SIGUCCS conference on User services: Mile high expectations*. New York, USA: ACM Press. pp.176-179

Price, J. (1984) *How to Write a Computer Manual: A Handbook of Software Documentation*. Menlo Park, California, USA: The Benjamin/Cummings Publishing Company

Quesenbery, W. (2001) *What Does Usability Mean: Looking Beyond "Ease of Use"*. [online] Available from: http://www.cognetics.com/presentations/whitney/more-than-ease-of-use.html [Accessed: 05/07/02]

Quesenbery, W. (2002) *The Five Dimensions of Usability.* [online] Available from: http://www.winwriters.com/ohc02/abstracts/abstract23.htm [Accessed: 09/07/02]

Ramey, J. (1989) Escher Effects in Online Text. In: E. Barrett, Edward (ed) *The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information.* Massachusetts, USA and London, UK: The MIT Press, pp. 388-402

Raskin, J. (2000) *The Humane Interface*, New York, USA: Addison-Wesley

Redish, J. C. (1988) Reading to learn to do. In: *Technical Writing Teacher*, Fall, 15 (3), 223-233

SAP AG (1997) *General Standards and Guidelines for Writing at SAP.* Walldorf, Germany: SAP

Schneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Massachusetts, USA: Addison Wesley Lingman

Schreurs, D. and Adriaens, G. (1992) Controlled English (CE): From COGRAM to ALCOGRAM. In: P. O' Brian Holt & N. Williams (eds) *Computers and Writing: State of the Art.* Oxford: Intellect Books

Schriver, K. A. (1997) *Dynamics in Document Design: Creating Texts for Readers.* New York, USA: John Wiley & Sons

Schwitter, R., Ljungberg, A. & Hood, D. (2003) ECOLE: A Look-ahead Editor for a Controlled language. In: *Proceedings of EAMT-CLAW 03: Controlled Language Translation.* Dublin, 15th-17th May, 2003. Geneva: EAMT, pp. 141-150

Sherif, M. (1937) An Experimental approach to the Study of Attitudes. In: *Sociometry*, 1, pp. 90-98

Simpson, M. (1990) How usability testing can aid the development of online documentation. In: *ACM SIGDOC Asterisk Journal of Computer Documentation, Proceedings of the 8th annual international conference on Systems documentation.* Vol. 14, Issue 4. pp. 41-47

Smart, K. L. & Whiting, M. E. (1994) Reassessing the documentation paradigm: writing for print and online. In: *ACM SIGDOC Proceedings of the 12th annual international conference on Systems documentation.* New York, USA: ACM Press. pp. 6-9

Sperber, D. & Wilson, D. (1986) *Relevance: communication and cognition.* Oxford: Blackwell

Sperber, D. & Wilson, D. (1995) *How Do We Communicate.* In: J. Brockman & K. Matson (eds) *How Things Are: A Science Toolkit for the Mind.* 191-199, New York, USA: Morrow

Spool, J. M. (1996) Smarter Usability Testing: Practical Techniques for Developing Products. In: *CHI96 ACM Conference Companion on Human Factors in Computing Systems: Common Ground.* New York, USA: ACM Press, pp. 365-366

Starr, M. (2001) RU31: Give them Printed Documentation, too!!! [online] Available from: http://www.tc-forum.org/topicus/ru31give.htm. [Accessed: 22/05/2004]

Sullivan, M. & Chapanis, A. (1983) "Human Factoring a Text Editor Manual". In: *Behaviour and Information Technology, 2, 2*, pp. 113-125

Teague, R., De Jesus, K. Nunes Ueno, M. (2001) Concurrent Vs. Post-Task Usability Test Ratings. In: *CHI '01 Extended Abstracts on Human Factors in Computing Systems.* pp. 289-290

Thovtrup, H. & Nielsen, J. (1991) Assessing the Usability of a User Interface Standard. In: *ACM Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology.* New York, USA: ACM Press. pp. 335-341

Tolman, E. (1948) Cognitive Maps in Rats and Men. In: *Psychological Review*, 55, pp.189-209

Toury, G. (1995) *Descriptive Translation Studies and Beyond.* Amsterdam, The Netherlands and Philadelphia, USA: John Benjamins.

Treisman, A. (1960) Contextual cues in selective listening. In: *Quarterly Journal of Experimental Psychology*, 12, pp. 242-248

Treisman, A. (1964) Strategies and models of selective attention. In: *Psychological Review.* 78, pp. 282-299

UsabilityNet (unknown:a) *Requirements Meeting.* [online] Available from: http://www.usabilitynet.org/tools/requirements.htm [Accessed: 24/02/03]

UsabilityNet (unknown:b) *Performance Testing Methods.* [online] Available from: http://www.usabilitynet.org/tools/testing.htm [Accessed: 24/02/03]

UsabilityNet (unknown:c) *Wizard of Oz Method.* [online] Available from: http://www.hostserver150.com/usabilit/tools/wizard.htm [Accessed: 24/02/03]

UsabilityNet (unknown:d) *Evaluate Prototype.* [online] Available from: http://www.hostserver150.com/usabilit/tools/evaluate.htm [Accessed: 24/02/03]

UsabilityNet (unknown:e) *International Standards.* [online] Available from: http://www.usabilitynet.org/tools/r_international.htm [Accessed: 24/02/03]

UsabilityNet (unknown:f) *Style Guides.* [online] Available from: http://www.usabilitynet.org/tools/style.htm [Accessed: 24/02/03]

UsabilityNet (unknown:g) *Scenarios of Use (Use Cases).* [online] Available from: http://www.usabilitynet.org/tools/scenarios.htm [Accessed: 24/02/03]

UsabilityNet (unknown:h) *Heuristic Evaluation.* [online] Available from: http://www.usabilitynet.org/tools/expertheuristic.htm [Accessed: 24/02/03]

UsabilityNet (unknown:i) *Task Analysis Methods.* [online] Available from: http://www.usabilitynet.org/tools/taskanalysis.htm [Accessed: 24/02/03]

UsabilityNet (unknown:j) *Definition of Usability.* [online] Available from: http://www.hostserver150.com/usabilit/management/b_standards.htm [Accessed: 24/02/03]

UsabilityNet (unknown:k) *Definition of Usability.* [online] Available from: http://www.hostserver150.com/usabilit/management/b_what.htm [Accessed: 24/02/03]

UsabilityNet (unknown:l) *Getting Started with User Centred Design.* [online] Available from: http://www.usabilitynet.org/tools/gettingstarted.htm [Accessed: 24/02/03]

UsabilityNet (unknown:m) *Key Principles of User-Centred Design.* [online] Available from: http://www.hostserver150.com/usabilit/management/b_design.htm [Accessed: 24/02/03]

Van Laan, K. and Julian, C (2001) *The Complete Idiot's Guide to Technical Writing.* Indianapolis, USA: Alpha Books

Vaus, D. A. de (1991) *Surveys in Social Research* (3rd Edition). London, UK: UCL Press

VDI 4500-2: 2000 *Technical documentation - Internal technical product documentation*"

Vermeer, H. J. (1978) Ein Rahmen für eine allgemeine Translationstheorie. In: *Lebende Sprachen.* Vol. 23, pp. 99-102.

Vygotsky, L.S. (1962) *Thought and Language* (E. Hanfmann & G. Vakar, Ed. & Trans.). Cambridge, Massachusetts, USA: The MIT Press

Wagner, J. (2001) *Mensch-Computer-Interaktion: Sprachwissenschaftliche Aspekte.* Frankfurt am Main, Germany: Peter Lang

Wammi Web Usabilty Questionnaire. [online] Available from: http://www.wammi.com/. [Accessed: 07/07/03]

Weiss, E. H. (1985) *How to Write a Usable User Manual.* Philadelphia, USA: ISI Press

Weiss, E. H. (1995) The Retreat from Usability: User Documentation in the Post-Usability Era. In: *ACM SIGDOC Asterisk Journal of Computer Documentation*, Volume 19, Issue 1, pp. 3-18

Wertheimer, M. (1959) *Productive Thinking.* New York, USA: Harper & Row

White, F. D. (1996) *Communicating Technology: Dynamic Processes and Models for Writers.* New York, USA: HarperCollins College Publishers

Wiklund, M. E. [ed] (1994) *Usability in Practice.* London, UK and New York, USA: Academic Press

Witkin, H. A., Oltman, P. T., Raskin, E. and Karp, S. A. (1971) *Group Embedded Figures Test Manual.* Palo Alto: Consulting Psychologists Press

Wixon, D. & Wilson, C. (1997) The usability engineering framework for product design and evaluation. In: M. Helander, T. K. Landauer, P. Prabhu (eds) *Handbook of Human-Computer Interaction.* 2nd edition. The Netherlands: Elsevier Science

Zirinsky, M. (1987) Usability Testing of Computer Documentation. In: *Proceedings of the 5th annual international conference on Systems documentation.* New York, USA: ACM Press, pp.61-65

User Guides Consulted:

Cylon Controls (1993) *UCC4 Programming Manual.* Dublin: Cylon Controls Ltd.

Cylon Controls (1994a) *Engineering Tool Manual.* Dublin: Cylon Controls Ltd.

Cylon Controls (1994b) *UC16 Programming Manual.* Dublin: Cylon Controls Ltd.

Cylon Controls (1994c) *UCC4 Keypad Manual.* Dublin: Cylon Controls Ltd.

Digitech (1997) *RP7 Valve User's Guide.* Utah, USA: Harman Music Group Incorporated

Informix (1997) *Informix Embedded SQL: TP/XA Programmer's Manual.* Menlo Park, CA: Informix Press

Lexmark (1998) *Lexmark Z51 User's Guide.* Lexington, USA: Lexmark International Inc.

APPENDICES

Appendix A
User Profile Questionnaire

User Profile Questionnaire

Name of Product:	
Name of Company / Organisation:	
Type of Company / Organisation:	
Number of Employees (where this product is used):	
Employee Job Title:	

Educational Background

1. What level of education is expected of employees?

☐ A. Degree ☐ B. Diploma ☐ C. Certificate ☐ D. Second Level ☐ E. Other

If you answered D or E, skip question 2.

2. What subjects were studied to obtain this qualification?

3. Which subjects previously studied were considered when recruiting employees?

Computer Skills

4. What level of computer skills are required for this job?

☐ A. Advanced ☐ B. Intermediate ☐ C. Beginners ☐ D. No Specific Requirements

5. Do employees typically have any specific computer qualifications prior to employent?

☐ A. Yes ☐ B. No

If yes, please give examples:

6. Do employees need these skills upon commencing employment?

☐ A. Yes ☐ B. No

7. What type of computer skills are needed for this job? Please tick all that apply.

☐ A. Word Processing ☐ B. Networking ☐ C. Email ☐ D. Databases

8. Any other computer skills?

Typing Speed

9. Is there a typing speed requirement for new employees?

☐ A. Yes ☐ B. No

10. If yes, what typing speed (in words per minute) is required?

☐ A. <20 wpm ☐ B. 20-40 wpm ☐ C. 40+ wpm

11. Please number the following skills in order of importance for potential employees?

☐ A. Typing Skills ☐ B. Computer Skills ☐ C. Language / Writing Skills

12. Are there any other skills which are vital for this job?

Appendix B
Main Study Task Sheet

Usability Study		Task Sheet
	Tester ID	P1
Page 1 of 1	Date	24/03/04

Task 1

First, you need to create two new entries in the QuicKey database to indicate the identity of each speaker: one for Ms. Smith and one for Mr. Jones.

Create the QuicKey entries as follows:

QuicKey Shortcut	Name
smith	Ms. Jane Smith:
jones	Mr. Michael Jones:

When you have finished, please inform the test administrator.

Task 2

Now start the DigiLog application and create a new log. We want to save the log automatically in **HTML** format so you will need to make the necessary settings in DigiLog.

Now set the working directory to: `C:\chamber`

When you have finished, please inform the test administrator.

Task 3

Now listen to the conversation between Ms. Smith and Mr. Jones. You don't have to type everything they say, just their name and the first 4-5 words they speak followed by **...** each time a person speaks.

Remember to type "Meeting" at the top of the page and hit return before you start.

Task 4

Although DigiLog has been saving the log in HTML format automatically as you typed, you also want to save another copy of the file manually in RTF format. Save the file in the following directory:

`C:/Usability Test/Pilot/p???/log1.rtf`

When you have finished, please inform the test administrator.

Appendix C
Transcript of Recorded Speech

Interview with The Rt Hon Chris Patten, CH (by BBC)

In Zagreb on Friday, 24th November 2000
Source: http://europa.eu.int/comm/external_relations/news/patten/bbc_24_11_00.htm

BBC Mr Patten, can you just characterise the symbolic significance of what is happening today.

PATTEN This year has been a much better one in the Balkans and South East Europe than any of us could have hoped for at the outset. First of all the democratic revolution in Croatia. Then the marvellously good news in Serbia a few weeks ago. And what this is all about is Europe living up to its promises. We told countries in the region that if they chose democracy, if they chose open economics, if they chose the rule of law, we would want to bring them closer to the European family. That is what we are doing with trade, with money, with political co-operation. And this Conference is putting the stamp on that process.

BBC There are still plenty of people, some here in Croatia, in Bosnia, in Kosovo, who are still fighting for nationalism and against democracy. Have they really been seen off?

PATTEN Nobody can pretend that all the problems are over. After all we have had some particularly unpleasant incidents in Kosovo in the last few days. But the problems of dealing with Balkans issues after Milosevic are a hell of a lot better than the problem of dealing with Milosevic. I think we have got a chance now of ending the awful, bloody decade in which the former Yugoslav state was dismembered with bloodshed, with refugees, with murder and mayhem. We have got a chance of giving South East Europe a properly European vocation.

BBC Do all these countries have to move forward together? I mean there is some worry here in Croatia that they might be held back from EU membership by others.

PATTEN No, we will deal with each country on its merits. Those are the agreements that we are signing with them. But part of our overall view is that just as in the European Union, we think that it makes good sense to try to be good neighbours; we think that it makes good sense to have a single market; we think that we have recovered from the two Wars of the last century, partly through political and economic integration. So we think the same about South East Europe. They should be good neighbours, they should trade with one another, but we will treat each of them as though it was on its own merits.

BBC Now in the midst of all this debate, your own Party in Britain is saying that the British Government is forcing British soldiers towards a European Army.

PATTEN I think the whole debate is quite extraordinary. I think it has been whipped up by some of the press who are hysterical in the weeks before Nice. Much of the reporting is fabricated or fatuous. But what is happening with the attempt for Europe to do more to look after its own defence interests is justified by this Conference. We are not undermining NATO. NATO has made it perfectly clear that it supports what European Ministers are trying to do. We are not undermining the Atlantic Alliance. The American administration has made it perfectly clear that it supports what we are trying to do. Nobody thinks for one moment that the Americans are going to reduce their strategic commitment to Europe. But there is plenty of evidence that if Europe doesn't do more for itself, the Americans will be less and less willing, more and more reluctant, to commit American lives for bloody, little European tragedies. And surely the history of Bosnia, the history of Kosovo, underlines the importance of Europe doing more for itself. Politicians, including Conservative politicians, have been arguing for that for years. Now we are trying to do it, and frankly it is crazy to suggest that this is the creation of a European Army or an attempt to kick the Americans out of Europe. Nothing could be further from the truth.

BBC But that is precisely what your own Party leader has been suggesting this week. Isn't it now time for you to consider leaving the Party. Others have done so.

PATTEN I am a Conservative. The Conservative Party has been the pro-European party for most of the last forty years. I disagree with William Hague on this. He knows perfectly well that I disagree with him about it, because I have written to him to say so. I think the Conservative Party has made a considerable mistake on this issue, and I hope that they won't be led on by some of the press who have become completely hysterical about it. Frankly on some front pages these days you see propaganda, not news reporting.

BBC Is the Conservative Party going to lose the argument in Britain, and lose any chance of getting your Party back into government?

PATTEN I am not interested in that. What I am interested in is what is in Britain's national interest. And what is in Britain's national interest is that we make sure that Europe does more to protect our interests and its interests. It is not just Britain that is involved in this. It is not just the countries of the European Union. There are 30 countries altogether who want to commit themselves to providing this Rapid Reaction Force. So are we saying that 30 countries are wrong, or that the other 29 are wrong, that the Americans are wrong. It is just not the case.

BBC Are Britain's relations with her European Union partners being damaged by this?

PATTEN I think what is damaging is that sometimes any attempt to be constructive in Europe is regarded by some of the media, by some newspapers as tantamount to treachery. I think that is profoundly against Britain's interests. It is perfectly possible to be patriotic and internationalist. To be patriotic and believe that Britain has a constructive role to play in helping to lead the European Union.

Appendix D
Task Event Log

Tester ID	P1
Date	

Time Measurements

Criterion	Instances		Total
1. Completion of Tasks	Task 1:	Task 2:	
	Task 3:	Task 4:	

Criteria Scores

Criterion	Instances	Total
1. Tasks Not Completed	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
2. Times user guide used	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
3. PTS Score	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
4. Incorrect icon choices	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
5. Incorrect menu choices	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
6. Verbal Interactions/Questions During Tasks	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
7. Observations of Frustration	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
8. Observations of Confusion	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
9. Observations of Satisfaction	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
10. Incorrect Commands / Input	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	
11. Stopped Work without Completing Task	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	

Appendix E
Usability Study Tester IDs

Usability Study	Tester Details & IDs
Page 1 of 2	Created:
	Updated:

1. [name]

Tester ID Code	P1
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

2. [name]

Tester ID Code	P2
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

3. [name]

Tester ID Code	P3
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

4. [name]

Tester ID Code	P4
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

5. [name]

Tester ID Code	P5
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

6. [name]

Tester ID Code	P6
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

7. [name]

Tester ID Code	P7
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

8. [name]

Tester ID Code	P8
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

9. [name]

Tester ID Code	P9
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

10. [name]

Tester ID Code	P10
Test Group	Control/Experimental
Email	
Signed Consent Form?	
Pre-Test Session (Date/Time)	
Test Session (Date/Time)	
Post-Test Session (Date/Time)	

Appendix F
Post-Test Survey Questions

1. Which of the following statements is true?

a. QuicKeys are enabled if there is a checkmark beside the relevant menu item
b. QuicKeys are enabled if the relevant menu item is highlighted
c. QuicKeys are enabled automatically when you type a semi-colon

2. What happens if you type a colon after a QuicKey shortcut?

a. DigiLog will crash
b. The full name appears
c. Nothing happens

3. What happens if you use a QuicKey shortcut that is not recognised?

a. An error message appears
b. *XXXXX* appears in the log text
c. QuicKey creates a new shortcut for you

4. How do you change the appearance of the text in a log?

a. Go to the *Edit* menu
b. Go to the *Print* menu
c. None of the above

5. How do you align text in a log along the left side of the page?

a. Open the *Paragraph* menu and then click on *Format*
b. Open the *Format* menu and click on *Paragraph*
c. Open the *Setup* menu and click on *Paragraph*

6. The Setup menu is used to access which of the following?

a. QuicKeys
b. Text Formatting
c. Save Format
d. Display/Hide Toolbar

7. How do you display the status bar?

a. Click on the *Show/Hide Status Bar* icon
b. Click on *Status Bar* in the Setup menu
c. Click on *Status Bar* in the View menu

8. How do you open an existing log?

a. Click on *Find* in the *File* menu
b. Click on *Open* in the *Setup* menu
c. Click on *Open* in *File* menu

9. How do you left align text in a log?

a. From the *View* menu, click on *Paragraph Format*
b. From the *Paragraph* menu, click on *Format* and then *Align Left*
c. From the *Format* menu, click on *Paragraph* and then *Align Left*

10. How do you paste text into a log?

11. What would you use the following toolbar button for?

a. To set the file format
b. To move a file to a different subdirectory
c. To change the size of text in a log

12. Which of the following are not menus in DigiLog?

a. File
b. Edit
c. Tools
d. View
e. Find

13. Which of the following file types *cannot* be created by DigiLog?

a. .HTML
b. .DOC
c. .RTF

14. Which of the following statements is false?

a. When you start DigiLog, a new log is automatically created
b. A new log is created when you go to the *File* menu and click *New*.
c. You create a new log by going to the *Setup* menu and clicking *New*.

15. How do you start DigiLog using the mouse?

Appendix G

Control Group User Guide – No Iconic Linkage

DiGiTAKE

Chapter
1

Chapter 1
DigiLog

Contents

1. Introduction

DigiLog is a fully featured word processing package that works in conjunction with the *DigiTake* digital audio recording package. *DigiTake* records audio as a series of small 10-15 minute files known as "takes". The purpose of *DigiLog* is to create a text log / annotation for each take in the system. It embeds in its text a time reference to the digital audio file and saves the information in several formats including RTF, text or HTML. This makes the application's output compatible with industry standard word processing packages such as Lotus Notes, Word, WordPerfect, etc. The application also allows the user to access a database of names, headings, etc. using QuickKey shortcuts. The new log is created automatically.

The DigiLog package contains the following features:

- QuicKey programming to speed entry of long names, titles etc.
- Auto header generation.
- Text formatting features
- Hardcopy printing.
- Pre-programmed Titles / Save Directories.
- Programmable QuicKey access to long names and titles.

2. Getting Started

DigiLog is used to create an electronic log or annotation of proceedings. It is similar in use to a basic word processing package and has a standard Windows style interface that all users will be familiar with. This means that there is no need for expensive proprietary software applications in order to review log files. They can be read using any standard word processing package. This section provides users with basic information to start using DigiLog.

How to Start DigiLog

Before commencing work with DigiLog, it is first necessary to open the application. This is done by going to the desktop and double-clicking the *DigiLog* icon. You can also start DigiLog by opening the *Start* menu and typing the following in the *Run* dialog:

```
C:/Usability Test/apps/digilog.exe
```

When DigiLog is started, a new log is automatically created.

Create a Log without QuicKeys

1. Disable *QuicKeys* by clicking *Use QuicKeys* from the *Setup* menu
2. *QuicKeys* are disabled when the check mark is removed from the *Use QuicKeys* menu item.
3. Type text in the log editor as in a conventional word processing editor.
4. Logs are changed and saved automatically.

Creating a Log using QuicKeys:

1. To enable *QuicKeys*, select *Use QuicKeys* from the *Setup* menu
2. If there is a checkmark before the *Use QuicKeys* menu item, QuicKeys are enabled.
3. To use a QuicKey, enter the pre-programmed *QuicKey* shortcut followed by a semi-colon. The corresponding name or title will appear in the editor.

> Note:
> After calling name or title using a *QuicKey* you MUST enter text before pressing enter. Pressing the enter key causes the text to be associated with the sound file. To ensure accurate association press enter immediately you have finished entering the text. Logs are created and saved automatically.

3.

File Menu

This section explains the various functions contained in the File menu.

New

- Click New to create a new blank document.

Open

1. Click Open to display the file open dialog box.
2. Select a log to open from the file open dialog box that appears.
3. Click the Open button in the dialog box to open the log.

Save

Click Save to manually save the current log.

Please Note:
Once the DigiLog system has stopped recording, all further edits thereafter must be saved manually by using the Save function.

Print

- Click Print to send the log to a printer.

Print Preview

- Click Print Preview to see a preview of what the log will look like when printed.

Restart

The restart button is used to restart the DigiLog system after the recording timer has been re-set. Following a break in proceedings, the restart button must be pressed after the recording package has been re-set.

1. Select *File* on the file menu bar.
2. Click the *Restart* button.

Selecting the *Restart* button will prompt the DigiTake system to display the name of the next take and the time the next take is set to begin. Please ensure that the take information displayed on screen at this time is correct.

If the take information is incorrect, the DigiTake system, and subsequently the DigiLog system, must be re-set to display the correct information.

Exit

- Click *Exit* to terminate DigiLog.

4. The Edit Menu

The *Edit* menu is used to edit the text which has been typed into the DigiLog log file.

Undo

- Click *Undo* to undo the last edit operation.

Cut

3. Highlight the text to cut.
4. Click *Cut* to cut the selected text from the log.

Copy

1. Highlight the text to copy.
2. Click *Copy* to copy the selected text to the clipboard.

Paste

1. Place the cursor at the point where the paste is to take place.
2. Click *Paste* to paste the contents of the clipboard to the log.

Select All

1. Click *Select All* to highlight all text in the log.
2. Highlighted text can be copied, cut, pasted or over-written.

Find

To find text in a log:

1. Click *Find*. The find dialog box will appear.
2. Enter the text you are looking for in the *Find What* field.
3. Click the *Find Next* button.

Find Next

- Click *Find Next* to find the next occurrence of a piece of text in a log.

Replace

1. Click *Replace* to open the Replace dialog box.
2. Enter the text to be replaced in the *Find What* field.
3. Enter the text to replace it in the *Replace What* field
4. To replace an individual occurrence of a piece of text, click the *Find Next* button until the occurrence is found.
5. Click the *Replace* button to replace the text with that entered in the Replace What field.
6. To replace all occurrences click the *Replace All* button.
7. Click *Cancel* to close the Replace dialog box.

5. Format Menu

In this section, the Format menu functions are explained.

Font

1. Click *Font* to display the font dialog box.
2. Select the desired font type, style, size, colour etc.
3. Click *OK* to effect the change.
4. Click *Cancel* to close without changing,.

Paragraph

To view the various paragraph formats:-

1. Click on *Format* on the menu bar.
2. Select paragraph
3. The three paragraph formats will appear.

A tick will appear beside the option you have selected.

Align Left

1. From the *File* menu, click Format and select Paragraph.
2. Click *Align Left*. The paragraphs will be aligned along the left of the page.

Align Centre

1. Click *Format* on the *file* menu, and select *Paragraph*.
2. Select *Align Centre*. Paragraphs will appear centred on the page.

Align Right

1. Go to the file menu and click *Format*. Now select *Paragraph*.
2. Select *Align Right*. The paragraphs in the log file will be aligned along the right edge of the page.

6. Setup Menu

From the Setup menu it is possible to configure various aspects of the Digilog system.

QuicKeys Setup

QuicKey is a type of database that allows administrators to pre-program frequently used names, terms, portfolio titles, procedural statements etc. for quick retrieval and automatic insertion into the log text as described earlier on. Rather than users typing in long names, terms or other phrases all that a user needs to do is enter a pre-defined shortcut.

Using QuicKeys in a log

- Enable QuicKeys by clicking *Use QuicKeys*.

The QuicKey option is enabled when there is a check mark in front of the Use QuicKeys option.

To disable QuicKeys in a log

- To disable QuicKeys, click *Use QuicKeys*.

If no checkmark appears in before the *Use QuicKeys* option, the QuicKey option is disabled.

Save Format

DigiLog can save a log in any of three formats

HTML	Select HTML to save in internet browser compatible format.
RTF	Choose the RTF option to save text and format information. RTF is compatible with most word processors.
TEXT	To save text without format information select Text .

To select, click the desired format. A check mark is placed before the current selection.

Printer Setup

- Click *Printer Setup* to display the Printer Setup dialog box.

7.

View Menu

The view menu allows you to view different types of information used by DigiLog.

QuicKey List

1. Click *QuicKey List* to display the list of programmed QuicKeys .
2. The list is displayed in two columns. The column on the left contains the name or title that will appear in the finished log. The column on the right contains the corresponding QuicKey.
3. Click the *Close* button to close the dialog box.

To enter a QuicKey:

Type the QuicKey letters listed on the right hand side followed by the semi-colon to call up the full QuicKey Name listed on the left hand side.

Note:
Before pressing enter, you MUST enter some text once you have called a name or title using a QuicKey.

When you press the enter key, the text is associated with the sound file. To ensure accurate association press enter immediately after you have finished entering the text.

Note:
When using QuicKeys, the semi-colon key can only be used to call up the entries in the QuicKey list. Using the semi-colon for an entry that is not in the QuicKey list will cause *XXXXXXXX*: to be entered in the text. This indicates that the QuicKey used is not valid.

Directory List

Click *Directory List* to display the list of predefined directories where logs can be saved. A directory from the Directory List must be selected in order to select a file path to which the logs will be created. Selecting a directory to save logs to:

1. Double-click the meeting title required from the left column.
2. The save/working directory will be set to save logs to file path in the right column.

You will be asked to confirm the path.

- Click *Yes* to confirm.
- To cancel the selection, click *No*.

Confirming *Yes* to the path will prompt Set Working Directory to appear highlighted on the bottom left of the dialog box. Select *Set Working Directory* if working directory is correct.

Three possible dialog boxes will appear at this point:

1. An alert sign saying that Sub Directory already exists.

This prompt tells the user that the DigiTake package also has the same file path and that the directory selected already exists.

Click *OK* to save both the logs created by the DigiLog package and the audio recordings created by the DigiTake package to the same folder.

2. An information dialog box will appear asking if the file path as shown in the dialog box is correct.

Example of file path: c:\Room_1\10_02_2003

Click *OK* to save the files to the file path specified in the dialog box.

3. A dialog box will appear stating Path to Date Sub_Directory not found.

Click *OK* and return to DigiTake package to create directory with correct path to date required.

Please Note: If error message below appears after option a) or b) as previously mentioned then the file paths for the DigiLog and DigiTake packages are not the same.

- Please check both *DigiLog* and *DigiTake* directory settings and network connections.
- Select *OK* if error message appears.
- Dialog box will appear immediately after error message stating that *WORKING DIRECTORY NOT SET*. This indicates an error condition.
- Please check for correct spelling of directory names and correct network communication.
- Immediately after the directory has been set stating the take information generated in the DigiTake package, a pink banner will appear across the screen if directory has been set correctly.
- Click *Cancel* to close without making a selection.

Toolbar

- Click *Toolbar* to hide or display the toolbar.

The Toolbar is enabled when there is a check mark in front of the Toolbar option and is disabled when there is no check mark in front of the Toolbar option.

Status Bar

- Click *Status Bar* to hide or display the Status bar.

The Status Bar is enabled when there is a check mark in front of the *Status Bar* option and is disabled when there is no check mark in front of the *Status Bar* option.

8. The DigiLog Toolbar

The toolbar contains a number of buttons which provide users with access to the main DigiLog functions without the need to access the applications menus. This lets you perform certain functions more quickly than if you were using the menus. The following paragraphs explain the individual buttons on the toolbar.

Figure 1: DigiLog Toolbar

Restart

Click on the *Restart* button after the recording timer has been re-set, following a break, to continue logging.

Open a new document

Select the *New* icon to open a new blank document.

Open an existing log

Click *Open* to open an existing log on your local drive or network.

Save

Click *Save* to manually save all logs and edits created after the DigiTake package has stopped recording.

Exit

Select the *Exit* button to close the DigiLog program.

Print Preview

Print Preview allows you to preview how each page will look before printing.

Print

Click on the *Print* icon to print the current document.

Cut

Selecting *Cut* allows you to cut the selected text from the log.

Copy

To copy the selected text to the clipboard, click *Copy*.

Paste

Select the *Paste* icon to paste the contents of the clipboard to the log.

View QuicKey List

Clicking on the *View QuicKey List* will allow you to view the list of QuicKeys unique to your system. By entering a value from the right-hand side of the list followed by the semi-colon, the corresponding value on the left-hand side will be inserted into the text.

Select Working Directory

Click *Working Directory* to display the list of programmed Save directories.

Format Font

Select the *Format Font* icon to change the font type, style, size, colour etc.

Bold

Click on the *Bold* icon to apply bold formatting to the selected text.

Italics

Select the *Italics* icon to apply italic formatting to the selected text.

Underline

Select the *Underline* icon to underline the selected text.

Align Left

Click on the *Align Left* icon to align the selected text to the left of the page.

Align Centre

Click on the *Align Centre* icon to align the selected text centre of the page.

Align Right

Click on the *Align Right* icon to align the selected text to the right of the page.

Product Info.

Click on *Product Info.* to display copyright and product version information.

Help

Selecting the *Help* icon displays help files regarding the DigiLog package.

Toolbar

The View menu contains an option which allows the toolbar to be seen or hiddon. This works via a toggle switch in the View menu.

To view the Toolbar:

1. Click on the *View* menu.
2. Select *Toolbar*.

The *View Toolbar* option is enabled when there is a check mark to the left of the *View Toolbar* option.

To hide the Toolbar:

1. Open the *View* menu.
2. Select *Toolbar*.

To hide the toolbar, make sure there is no checkmark beside the Toolbar option.

Status Bar

The Status Bar is the long grey bar along the bottom of your screen as you operate DigiLog. By hovering your mouse over the icons on the toolbar, a description of their function will appear on the status bar.

There is an option within the View menu allowing the Status Bar to be seen or hidden from view.

To view the Status Bar:

1. Click on the *View* menu.
2. Select *Status Bar*.

If there is a check mark to the left of the view Status Bar option, the status bar will be displayed.

To hide the Status Bar:

1. Go to the *View* menu.
2. Select *Status Bar*.

The *View Status Bar* option is disabled when there is no check mark to the left of the view Status Bar option.

DiGiTAKE

Chapter 2
QuicKey

9. QuicKeys Setup

QuicKey is a database module that facilitates the programming of frequently used names, terms, portfolio titles, procedural statements etc. to allow the quick retrieval and automatic insertion into the log text. Instead of typing in long names, terms or other phrases, all users need to do is enter a pre-defined shortcut. To open QuicKey, use the *QuicKey* icon on the desktop. You can also start QuicKey by opening the *DigiLog* folder in the *Start* menu.

Add Entries to the QuicKey Database

The QuicKey database can be customised to allow the addition of new QuicKey shortcuts. This is done as follows:

Select QuicKey Setup from the Setup menu.

1. In the *Name* field, enter the name to be added to the database.
2. Click *Find*. A dialog box stating that the record does not exist appears and users are given the option of adding the new record. Click *Yes* to create the record.
3. Enter the name and QuicKey shortcut which will call up the name and click *Save*.

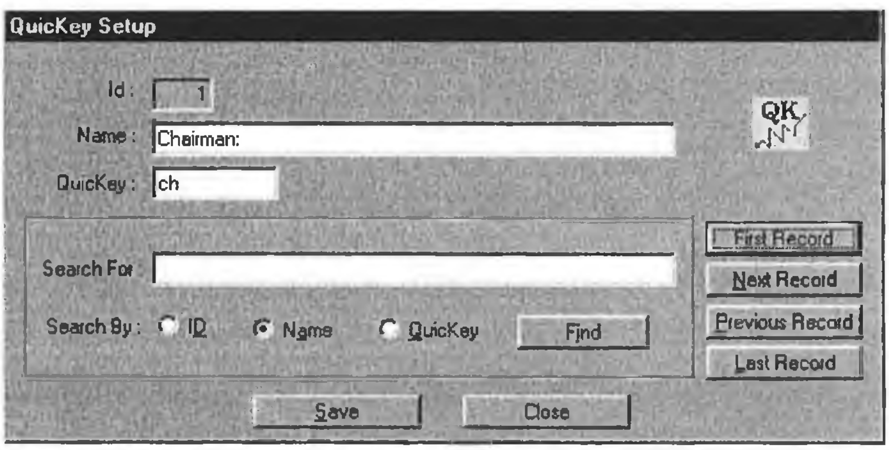


Figure 2: The QuicKey Setup Dialog

Important:
DigiLog only reads QuicKeys from the database when you first start DigiLog. To use the new QuicKey entries in a log, you must first restart DigiLog.

To Use QuicKeys in a Log

- QuicKeys are enabled by clicking *Use QuicKeys*.

If there is a checkmark in front of the *Use QuicKeys* option, the QuicKey option is enabled.

Disabling QuicKeys in a Log

- When you click Use *QuicKeys*, QuicKeys are disabled.

If there is no checkmark before the *Use QuicKeys* option, the QuicKey option is disabled.

10. Index

Appendix H

Experimental Group User Guide – Iconic Linkage Present

DIGITAKE

Chapter
1

Chapter 1
DigiLog

DIGITAKE

Contents

1. Introduction

DigiLog is a fully featured word processing package that works in conjunction with the *DigiTake* digital audio recording package. *DigiTake* records audio as a series of small 10-15 minute files known as "takes". The purpose of *DigiLog* is to create a text log / annotation for each take in the system. It embeds in its text a time reference to the digital audio file and saves the information in several formats including RTF, text or HTML. This makes the application's output compatible with industry standard word processing packages such as Lotus Notes, Word and WordPerfect. The application also allows you to access a database of names, headings and other words using QuickKey shortcuts. The new log is created automatically.

DigiLog contains the following features:

- QuicKey programming to speed entry of long names, titles etc.
- Auto header generation.
- Text formatting features
- Hardcopy printing.
- Pre-programmed Titles / Save Directories.
- Programmable QuicKey access to long names and titles.

2. Getting Started

DigiLog is used to create an electronic log or annotation of proceedings. It is similar in use to a basic word processing package and has a standard Windows style interface that all users will be familiar with.This means that there is no need for expensive proprietary software applications in order to review log files. They can be read using any standard word processing package. This section shows you how to start using DigiLog.

Starting DigiLog

To begin working, you will first need to start DigiLog. To do this, double-click the *DigiLog* icon on the desktop. You can also start DigiLog by opening the *Start* menu and typing the following in the *Run* dialog:

```
C:/Usability Test/apps/digilog.exe
```

When DigiLog is started, a new log is automatically created.

Creating a Log without QuicKeys

1. To disable *QuicKeys*, go to the *Setup* menu and click *Use QuicKeys*.
2. *QuicKeys* are disabled if there is no check mark beside the *Use QuicKeys* menu item.
3. Type text in the log editor as you would using a conventional word processing editor.
4. Logs are changed and saved automatically.

Creating a Log using QuicKeys

1. To enable *QuicKeys*, go to the *Setup* menu and click *Use QuicKeys*.
2. *QuicKeys* are enabled if there is a check mark beside the *Use QuicKeys* menu item.
3. To use a QuicKey, enter the pre-programmed *QuicKey* shortcut followed by a semi-colon. The corresponding name or title appears in the editor.

> Note:
> If you use a *QuicKey* to insert a name or title, you MUST enter text before pressing enter. When you press the enter key, the text is associated with the sound file. To ensure accurate association, press enter immediately after you finish typing the text. Logs are changed and saved automatically.

3. The File Menu

This section explains the various functions contained in the File menu.

New

- To create a new blank document, click *New*.

Open

1. To open a file, click *Open*.
2. The File Open dialog box appears. Select the log you want to open.
3. To open the log, click *Open*.

Save

To manually save the current log, click *Save*.

> Note:
> If the DigiLog system has stopped recording, you must manually save all subsequent changes using the Save function.

Print

- To send the log to a printer, click *Print*.

Print Preview

- To see what the log will look like when printed, click *Print Preview*.

Restart

The *Restart* button is used to restart the DigiLog system after the recording timer has been re-set. If there has been a break in proceedings, press the *Restart* button after the recording package has been re-set.

1. From the *File* menu, select *File*.
2. Click *Restart*.

When you click on Restart, the DigiTake system displays the name of the next take and the time the next take is set to begin. Make sure that the take information displayed on screen at this time is correct.

If the take information is incorrect, reset the DigiTake system and the DigiLog system in order to display the correct information.

Exit

- To close DigiLog, click *Exit*.

4. The Edit Menu

The Edit menu is used to edit the text which has been typed into the DigiLog log file.

Undo

- To reverse the last edit operation, click *Undo*.

Cut

1. Highlight the text you want to cut.
2. To cut the selected text from the log, click *Cut*.

Copy

1. Highlight the text you want to copy.
2. To copy the selected text to the clipboard, click *Copy*.

Paste

1. Place the cursor at the point where you want to paste the text.
2. To paste the contents of the clipboard into the log, click *Paste*.

Select All

1. To highlight all of the text in a log, click *Select All*.
2. Once you have highlighted the text, you can copy, cut, paste or over-write it.

Find

To find text in a log:

1. Click *Find*. The *Find* dialog box appears.
2. Enter the text you are looking for in the *Find What* field.
3. Click *Find Next*.

Find Next

- To find the next occurrence of a piece of text in a log, click *Find Next*.

Replace

1. To open the Replace dialog box, click *Replace.*
2. Enter the text you want to replace in the *Find What* field.
3. Enter the new text you want to add in the *Replace What* field.
4. To replace an individual occurrence of a piece of text, click *Find Next* until you find the next occurence.
5. To replace the text with the new text you entered in the *Replace What* field, click *Replace.*
6. To replace all occurrences, click *Replace All.*
7. To close the *Replace* dialog box, click *Cancel.*

5. The Format Menu

This section explains the various *Format* menu functions.

Font

1. To display the font dialog box, click *Font*.
2. Select the desired font type, style, size and colour.
3. To apply your changes, click *OK*.
4. To close the *Font* dialog box without applying your changes, click *Cancel*.

Paragraph

To view the various paragraph formats:

1. From the *format* menu, select *Paragraph*.
2. The three paragraphs formats appear.

A check mark indicates which option you have selected.

Align Left

1. From the *File* menu, click *Format* and select *Paragraph*.
2. Select *Align Left*. The paragraphs are now aligned along the left of the page.

Align Centre

1. From the *File* menu, click *Format* and select *Paragraph*.
2. Select *Align Centre*. The paragraphs are now centred on the page.

Align Right

1. From the *File* menu, click *Format* and select *Paragraph*.
2. Select *Align Right*. The paragraphs are now aligned along the right of the page.

6. The Setup Menu

From the Setup menu you can configure various aspects of the Digilog system.

QuicKeys Setup

QuicKey is a type of database that allows administrators to pre-program frequently used names, terms, portfolio titles, procedural statements, for example, so that they can be retrieved quickly and inserted automatically into the log text. Instead of typing in long names, terms or other phrases, all all you need to do is enter a pre-defined shortcut.

Using QuicKeys in a log

- To enable *QuicKeys*, go to the Setup menu and click *Use QuicKeys*.

The QuicKey option is enabled when there is a check mark in front of the Use QuicKeys option.

Disabling QuicKeys in a log

- To disable *QuicKeys*, go to the Setup menu and click *Use QuicKeys*.

The QuicKey option is disabled when there is no check mark in front of the Use QuicKeys option.

Save Format

DigiLog can save a log in any of three formats

HTML	To save in internet browser compatible format, select *HTML*.
RTF	To save text and format information, select *RTF*. RTF is compatible with most word processors.
TEXT	To save text without format information, select *TEXT*.

Simply select the format you want to use. A check mark appears beside the current selection.

Printer Setup

- To display the Printer Setup dialog box, click Printer Setup.

7. The View Menu

The view menu allows you to view different types of information used by DigiLog.

QuicKey List

1. To display the list of pre-programmed QuicKeys, click *QuicKey List*.
2. The list is displayed in two columns. The column on the left contains the name or title that will appear in the finished log. The column on the right contains the corresponding QuicKey.
3. To close the dialog box, click *Close.*

Inserting a QuicKey:

To insert a full QuicKey Name, type the corresponding QuicKey shortcut followed by a semi-colon.

> Note:
> If you use a *QuicKey* to insert a name or title, you MUST enter text before pressing enter.

When you press the enter key, the text is associated with the sound file. To ensure accurate association, press enter immediately after you finish typing the text.

> Note:
> When using QuicKeys, you can only use the semi-colon key to call up the entries in the QuicKey list. If you use the semi-colon for an entry that is not in the QuicKey list, *XXXXXXXX*: will appear in the log text. This indicates that the QuicKey used is not valid.

Directory List

To display the list of pre-programmed directories where you can save logs, click *Directory List.* To specify a file location to save the logs, you must select a directory from the Directory List. To select the directory you want to save logs in, do the following:

1. Double-click the meeting title required from the column on the left.
2. The save/working directory will be set to save logs to the file path in the right column.

You are now asked to confirm the path.

- To confirm the path, click *Yes*.
- To cancel the path, click No.

If you click *Yes*, *Set Working Directory* is highlighted on the bottom left of the dialog box. If the working directory is correct, click *Set Working Directory*.

One of the following dialog boxes appears at this point:

1. Sub Directory Already Exists.

This prompt tells you that the DigiTake package also has the same file path and that the directory selected already exists.

To save the logs created by the DigiLog package and the audio recordings created by the DigiTake package in the same folder, click *OK*.

2. Confirm File Path.

Example of file path: c:\Room_1\10_02_2003

To save the files to the location specified in the dialog box, click *OK*.

3. Path to Date Sub_Directory Not Found.

To create a directory with the correct path to date required, click *OK* and return to the DigiTake package.

Note:
If an error message appears after option 1 or 2, the file paths for the DigiLog and DigiTake packages are not the same.

- Check the directory settings and network connections for *DigiLog* and *DigiTake*.
- If an error message appears, click *OK*.
- A dialog box will appear immediately after error message stating that *WORKING DIRECTORY NOT SET*. This indicates that an error has occurred.
- Check that directory names have been spelled correctly and that there are no network communication problems.
- If the directory has been set correctly, a pink banner containing the take information generated in the DigiTake package appears on immediately after the you set the directory.
- To close the dialog box without applying your changes, click *Cancel*.

Toolbar

- To hide or display the toolbar, click *Toolbar*.

If there is a check mark beside *View Toolbar*, the toolbar is displayed. If there is no check mark beside *View Toolbar*, the toolbar is not displayed.

Status Bar

- To hide or display the Status Bar, click *Status Bar*.

If there is a check mark beside *View Status Bar*, the status bar is displayed. If there is no check mark beside *View Status Bar*, the status bar is not displayed.

8. The DigiLog Toolbar

The toolbar contains a number of buttons which let you access the main DigiLog functions without having to go through the various menus. This lets you perform certain functions more quickly than if you were using the menus. The following paragraphs explain the individual buttons on the toolbar.

Figure 1: The DigiLog Toolbar

Restart

If the recording timer has been re-set following a break, click *Restart* to continue logging.

Open a new document

To create a new blank document, click *New*.

Open an existing log

To open an existing log on your local drive or network, click *Open*.

Save

To manually save all logs and edits created after the DigiTake package has stopped recording, click *Save*.

Exit

To close the DigiLog program, click *Exit*.

Print Preview

To preview how each page will look before you print it, click *Print Preview*.

Print

To print the current document, click *Print*.

Cut

To cut the selected text from the log, click *Cut*.

Copy

To copy the selected text to the clipboard, click *Copy*.

Paste

To paste the contents of the clipboard into the log, click *Paste*.

View QuicKey List

To view the list of QuicKeys unique to your system, click on *View QuicKey List*. When you enter a value shown on the right-hand side of the list followed by the semi-colon, the corresponding value on the left-hand side is inserted into the text.

Select Working Directory

To display the list of pre-programmed directories where you can save logs, click *Working Directory*.

Format Font

To change the font type, style, size and colour of the text, click *Format Font*.

Bold

To embolden the selected text, click *Bold*.

Italics

To italicise the selected text, click *Italics*.

Underline

To underline the selected text, click *Underline*.

Align Left

To align the selected text to the left of the page, click *Align Left*.

Align Centre

To centre the selected text on the page, click *Align Centre*.

Align Right

To align the selected text to the right of the page, click *Align Right*.

Product Info.

To display copyright and product version information, click *Product Info*.

Help

To get help on using DigiLog, click *Help*.

Toolbar

Using the *View* menu you can either display or hide the toolbar.

Displaying the Toolbar:

- From the *View* menu, select *Toolbar*.

If there is a check mark beside *View Toolbar*, the toolbar is displayed.

Hiding the Toolbar:

- From the *View* menu, select *Toolbar.*

If there is no check mark beside *View Toolbar*, the toolbar is not displayed.

Status Bar

The Status Bar is the long grey bar along the bottom of your screen as you operate DigiLog. By hovering your mouse over the icons on the toolbar, a description of their function appears on the status bar.

Using the *View* menu you can either display or hide the status bar.

Displaying the the Status Bar:

- From the *View* menu, select *Status Bar*.

If there is a check mark beside *View Status Bar*, the status bar is displayed.

Hiding the the Status Bar:

- From the *View* menu, select *Status Bar*.

If there is no check mark beside *View Status Bar*, the status bar is not displayed.

DiGiTAKE

Chapter
2

Chapter 2
QuicKey

9. QuicKeys Setup

QuicKey is a type of database that allows administrators to pre-program frequently used names, terms, portfolio titles, procedural statements, for example, so that they can be retrieved quickly and inserted automatically into the log text. Instead of typing in long names, terms or other phrases, all all you need to do is enter a pre-defined shortcut. To open QuicKey, go to the *Start* menu and open the *DigiLog* folder. You can also start QuicKey by double-clicking the *QuicKey* icon on the desktop.

Adding Entries to the QuicKey Database

You can customise the QuicKey database to add new QuicKey shortcuts. You can do this as follows:

From the *Edit* menu, select *QuicKeys*.

1. In the *Search For* field, enter the name you want to add to the database.
2. Now click *Find*. A dialog box appears telling you that the name does not exist and you are asked whether you want to create a new record. To create the new record, click *Yes*.
3. In the QuicKey field, enter the QuicKey shortcut which will call up the name. Now click *Save*.

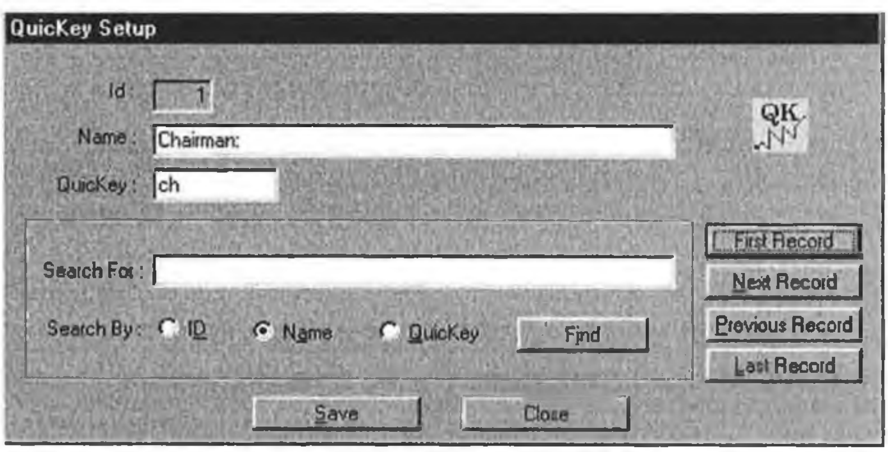


Figure 2: The QuicKey Setup Dialog

> Important:
> DigiLog only reads QuicKeys from the database when you first start DigiLog. To use the new QuicKey entries in a log, you must first restart DigiLog.

Using QuicKeys in a Log

- To enable QuicKeys in a log, go to the Setup menu and click *Use QuicKeys*.

The QuicKey option is enabled when there is a check mark in front of the Use QuicKeys option.

Disabling QuicKeys in a Log

- To disable QuicKeys in a log, go to the Setup menu and click *Use QuicKeys*.

The QuicKey option is disabled when there is no check mark in front of the Use QuicKeys option.

10. Index

Appendix I
Usability Study Consent Form

1. THE PURPOSE OF THIS PROJECT

The purpose of this study is to examine a piece of software and its user guide. By participating in this study as a tester you will help us find ways of making software and user guides easier to use.

2. YOUR ROLE AS A TESTER

As a tester you will test the user guide to find problems and identify areas where we can make improvements. It is important to remember that you are not being tested – instead you will be testing the product and its user guide. If something is not clear, difficult or does not work, it means that there is a problem with the product and not with your abilities.

3. WHAT WILL HAPPEN?

The study will take place in a special usability laboratory in Leinster House where you will be observed using the product and user guide. You will be asked to spend 20 minutes familiarising yourself with the user guide before you perform a number of tasks using the software. When you have completed these tasks we will ask you a number of questions to find out your thoughts, opinions and feelings with regard to the user guide. One week after the test we will ask you to complete a short test to determine how well the user guide explained the information.

4. WHAT INFORMATION WILL BE COLLECTED?

We will record information about how you use the product and the user guide. We will ask you to perform tasks and answer a series of questions in the form of a questionnaire. We will also record what happens on the computer screen as you perform tasks. In addition, all or some of the test will be videotaped. The purpose of this is to allow us to analyse the test in more detail.

5. YOUR ANONYMITY AND PRIVACY

During the study you will be assigned a unique ID number. All information you provide, including documents, questionnaires, files and video tapes will be identified with this ID number only. With the exception of the test administrator, nobody will know your identity or be able to link your name to any of the data. The information you provide will be used in a PhD thesis and may also be used in research papers to be published in refereed academic journals. However, your anonymity will be assured and your identity will never be revealed.

6. CONFIDENTIALITY

During the course of this study you will be using proprietary, copyright software which is supplied by a private company. Due to the commercial sensitivity of information relating to this software, you must not discuss the product or disclose any details relating to it. All documents and manuals must be returned to the test administrator before you leave the usability lab. You must delete all computer files relating to the product and return and print-outs made from these files.

7. YOUR COMFORT DURING THE STUDY

We will provide refreshments during the test and you may take a break at any time. Simply inform the test administrator that you would like to take a break. You are free to withdraw from the study at any time. If you have any questions regarding the study, please do not hesitate to ask the test administrator.

If you agree with these terms, please indicate your acceptance by signing below.

Name:	Test Administrator:
Signature:	Signature:
Date:	Date:

Appendix J
QUIS Post-Task Survey

PART 1: System Experience

1.1 How long have you worked with PCs?

__ less than 6 months

__ 6 months to less than 1 year

__ 1 year to less than 2 years

__ 2 years to less than 3 years

__ 3 years or more

1.2 On average, how much time do you spend per week using a PC?

	__ less than one hour	__ 4 to less than 10 hours
	__ one to less than 4 hours	__ over 10 hours

PART 2: Past Experience

2.1 How many operating systems have you worked with?

__ none	__ 3-4
__ 1	__ 5-6
__ 2	__ more than 6

2.2 Of the following devices, software, and systems, check those that you have personally used and are familiar with:

__ computer terminal	__ track ball	__ spreadsheet software
__ personal computer	__ joystick	__ database software
__ lap top computer	__ pen based computing	__ computer games
__ color monitor	__ graphics tablet	__ voice recognition
__ touch screen	__ head mounted display	__ video editing systems
__ floppy drive	__ modems	__ CAD applications
__ CD-ROM drive	__ scanners	__ rapid prototyping systems
__ keyboard	__ word processor	__ e-mail
__ mouse	__ graphics software	__ internet

PART 3: Overall User Reactions

Please circle the numbers which most appropriately reflect your impressions about using this computer system.
Not Applicable = NA.

3.1 Overall reactions to the system:	terrible wonderful 1 2 3 4 5 6 7 8 9	NA
3.2	frustrating satisfying 1 2 3 4 5 6 7 8 9	NA
3.3	dull stimulating 1 2 3 4 5 6 7 8 9	NA
3.4	difficult easy 1 2 3 4 5 6 7 8 9	NA
3.5	rigid flexible 1 2 3 4 5 6 7 8 9	NA

PART 4: Page Design

4.1	**Page layout was helpful**	never always 1 2 3 4 5 6 7 8 9 NA
4.1.1	Amount of information displayed on a page	inadequate adequate 1 2 3 4 5 6 7 8 9 NA
4.1.2	Arrangement of information on page	illogical logical 1 2 3 4 5 6 7 8 9 NA

Please write your comments about the page design and layout here:

PART 5: Terminology

5.1	**Use of terminology throughout system**	inconsistent consistent 1 2 3 4 5 6 7 8 9 NA
5.1.1	Work related terminology	inconsistent consistent 1 2 3 4 5 6 7 8 9 NA
5.1.2	Computer terminology	inconsistent consistent 1 2 3 4 5 6 7 8 9 NA
5.2	**Terminology relates well to the work you are doing?**	never always 1 2 3 4 5 6 7 8 9 NA
5.2.1	Computer terminology is used	too frequently appropriately 1 2 3 4 5 6 7 8 9 NA

Please write your comments about terminology here:

PART 6: User Guide

6.1	**The user guide is**	confusing 1 2 3 4 5 6 7 8 9 clear NA
6.1.1	The terminology used in the user guide	confusing 1 2 3 4 5 6 7 8 9 clear NA
6.2	**Language used in the user guide is consistent**	never 1 2 3 4 5 6 7 8 9 always NA
6.3	**Understanding information in the user guide is**	difficult 1 2 3 4 5 6 7 8 9 easy NA
6.3.1	Finding a solution to a problem using the user guide	impossible 1 2 3 4 5 6 7 8 9 easy NA
6.4	**Amount of help given**	inadequate 1 2 3 4 5 6 7 8 9 adequate NA
6.4.1	User guide defines specific aspects of the system	inadequately 1 2 3 4 5 6 7 8 9 adequately NA
6.4.2	Finding specific information using the user guide	difficult 1 2 3 4 5 6 7 8 9 easy NA
6.5	**Instructions for performing tasks are clear and unambiguous**	never 1 2 3 4 5 6 7 8 9 always NA

Please write your comments about the user guide here:

PART 7: User Guide Content & Structure

7.1	**The user guide was**	useless 1 2 3 4 5 6 7 8 9 helpful NA
7.1.1	The user guide is meaningfully structured	never 1 2 3 4 5 6 7 8 9 always NA
7.2	**The content of the user guide was**	useless 1 2 3 4 5 6 7 8 9 helpful NA
7.2.1	Information for specific aspects of the system were complete and informative	never 1 2 3 4 5 6 7 8 9 always NA
7.2.2	Information was concise and to the point	never 1 2 3 4 5 6 7 8 9 always NA
7.3	**Tasks can be completed**	with difficulty 1 2 3 4 5 6 7 8 9 easily NA
7.3.1	Instructions given for completing tasks	confusing 1 2 3 4 5 6 7 8 9 clear NA
7.3.2	Time given to perform tasks	inadequate 1 2 3 4 5 6 7 8 9 adequate NA
7.4	**Learning to operate the system using the user guide was**	difficult 1 2 3 4 5 6 7 8 9 easy NA
7.4.1	Completing system tasks after using only the user guide was	difficult 1 2 3 4 5 6 7 8 9 easy NA

Please write your comments about the structure and content of the user guide here:

PART 8: Learning

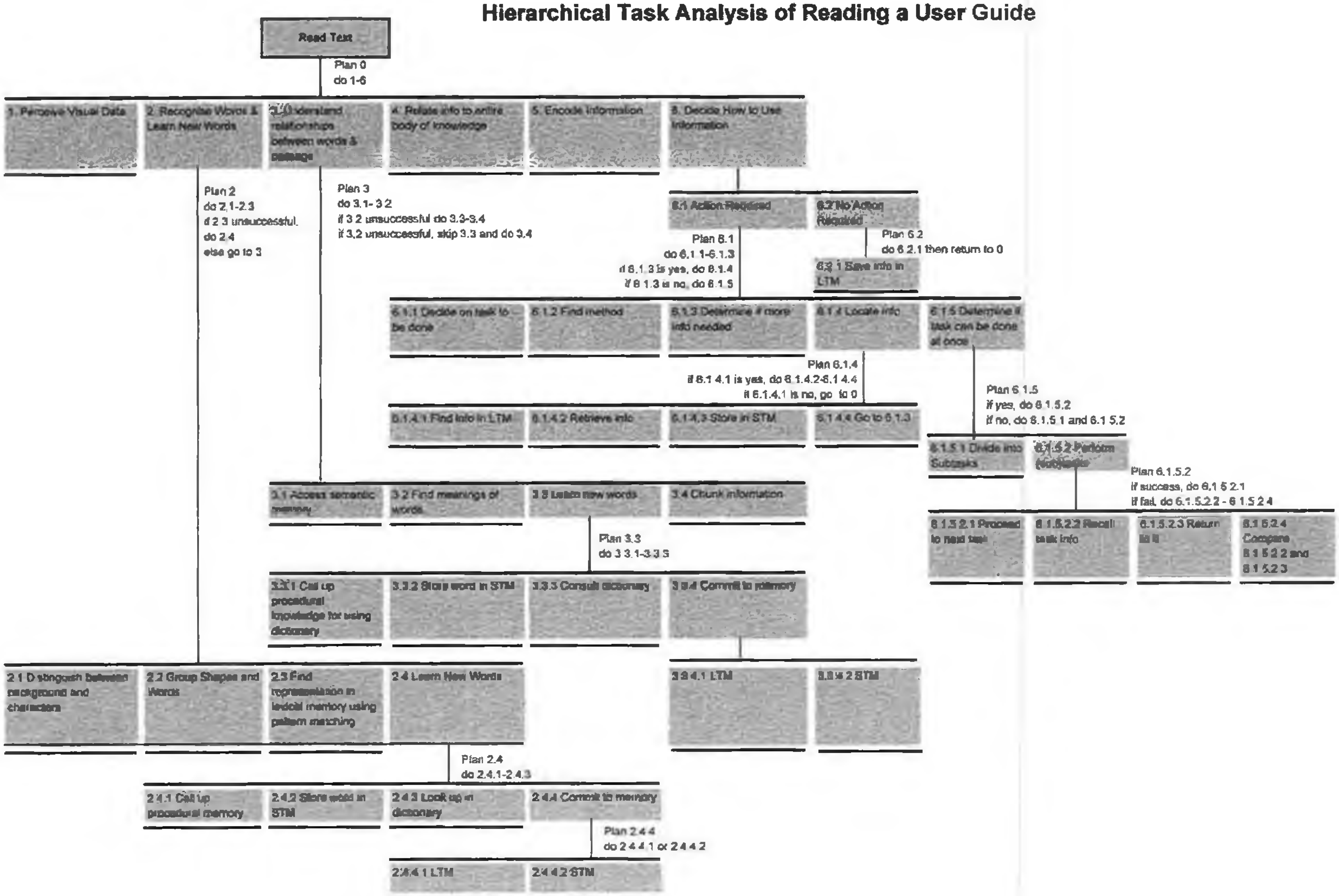
8.1	**Learning to operate the system**	difficult 1 2 3 4 5 6 7 8 9 easy NA
8.1.1	Getting started	difficult 1 2 3 4 5 6 7 8 9 easy NA
8.1.2	Learning advanced features	difficult 1 2 3 4 5 6 7 8 9 easy NA
8.1.3	Time to learn to use the system	slow 1 2 3 4 5 6 7 8 9 fast NA
8.2	**Exploration of features by trial and error**	discouraging 1 2 3 4 5 6 7 8 9 encouraging NA
8.3	**Remembering names and use of commands**	difficult 1 2 3 4 5 6 7 8 9 easy NA
8.4	**Tasks can be performed in a straight-forward manner**	never 1 2 3 4 5 6 7 8 9 always NA
8.4.1	Number of steps per task	too many 1 2 3 4 5 6 7 8 9 just right NA
8.4.2	Steps to complete a task follow a logical sequence	never 1 2 3 4 5 6 7 8 9 always NA

Please write your comments about learning to use the system here:

Appendix K

Hierarchical Task Analysis of Reading a User Guide

Hierarchical Task Analysis of Reading a User Guide



Appendix L

Individual Pilot Study Data from QUIS Questionnaire

	Control Group			Experimental Group		
Question	*P1*	*P3*	*P5*	*P2*	*P4*	*P6*
3.1	4	4	4	3	7	5
3.2	4	3	3	4	5	4
3.3	4	5	6	5	5	4
3.4	8	3	4	5	7	4
3.5	5	5	6	6	5	6
4.1	7	4	7	4	7	8
4.11	6	2	7	4	9	8
4.12	7	2	7	6	7	7
5.1	8	2	7	8	7	7
5.1.1	0	4	7	8	7	8
5.1.2	8	5	7	8	7	8
5.2	0	4	6	8	7	7
5.2.1	7	6	6	8	9	9
6.1	7	1	4	7	8	7
6.1.1	7	4	6	8	8	9
6.2	6	3	6	8	8	8
6.3	8	5	4	7	8	6
6.3.1	5	2	4	5	5	7
6.4	7	3	4	6	8	7
6.4.1	7	2	3	7	8	8
6.4.2	9	2	5	7	5	9
6.5	6	8	4	7	5	8
7.1	8	4	8	7	9	7
7.1.1	8	5	7	7	7	9
7.2	8	5	6	7	8	7
7.2.1	8	2	2	4	8	7
7.2.2	8	2	6	7	9	9
7.3	5	1	5	6	8	6
7.3.1	7	1	4	7	9	7
7.3.2	9	5	9	9	9	9
7.4	8	2	4	7	8	6
7.4.1	8	1	4	5	8	5
8.1	8	5	4	5	8	8
8.1.1	8	3	6	6	9	6
8.1.2	8	3	4	4	7	8
8.1.3	8	5	5	5	8	8
8.2	8	3	6	9	8	9
8.3	8	2	7	4	8	7
8.4	8	3	6	6	8	8
8.4.1	8	3	5	9	8	7
8.4.2	8	2	3	7	9	8
Average Rating	**6.804**	**3.317**	**5.317**	**6.341**	**7.512**	**7.195**

Group Average: 5.146 Group Average: 7.016

Appendix M

Individual Main Study Data from QUIS Questionnaire

Question	Control Group P1	P3	P5	P7	P9	Experimental Group P2	P4	P6	P8	P10
3.1	6	5	5	5	4	5	6	6	7	6
3.2	5	1	5	3	3	4	7	6	7	7
3.3	6	5	7	6	6	7	6	8	6	5
3.4	5	1	5	4	4	4	5	6	7	5
3.5	3	2	5	3	6	7	7	4	5	6
4.1	6	1	8	6	7	6	4	6	8	4
4.11	7	1	5	7	7	7	6	3	8	4
4.12	6	1	9	7	7	5	6	6	8	6
5.1	6	5	7	7	5	9	7	9	6	8
5.1.1	6	5	5	7	7	7	6	8	8	7
5.1.2	7	7	5	7	7	9	9	8	6	8
5.2	9	7	5	6	6	8	9	7	8	8
5.2.1	4	1	6	7	6	8	5	7	7	8
6.1	7	1	5	6	4	5	8	5	8	7
6.1.1	4	1	7	3	6	4	5	7	7	8
6.2	6	5	7	7	6	7	9	7	8	8
6.3	6	1	5	5	4	6	7	7	9	7
6.3.1	9	3	4	5	4	5	7	7	6	5
6.4	6	1	7	8	4	8	9	4	8	6
6.4.1	7	2	5	4	3	7	8	6	7	7
6.4.2	7	3	4	8	5	8	7	7	6	7
6.5	3	9	3	6	4	7	8	6	3	7
7.1	9	2	6	6	8	7	7	7	8	8
7.1.1	3	3	7	7	7	7	4	6	7	7
7.2	9	2	7	7	6	6	6	7	9	7
7.2.1	5	3	4	7	2	5	8	7	8	4
7.2.2	7	2	5	9	6	6	9	8	8	7
7.3	5	1	3	6	5	5	7	8	5	6
7.3.1	9	1	8	5	4	7	7	7	6	7
7.3.2	7	6	5	5	9	7	9	6	9	9
7.4	4	1	5	6	4	5	9	7	9	7
7.4.1	3	6	4	4	4	5	7	8	7	5
8.1	5	3	5	6	4	6	9	8	8	5
8.1.1	9	2	4	6	6	7	7	8	8	6
8.1.2	4	1	5	4	4	6	8	7	5	5
8.1.3	4	2	3	6	5	6	9	7	5	5
8.2	2	5	5	7	6	4	9	4	5	9
8.3	4	7	6	7	7	6	6	8	6	5
8.4	6	3	5	7	6	6	6	7	7	6
8.4.1	9	1	5	7	5	7	7	7	6	9
8.4.2	4	1	5	7	3	7	4	8	7	7
Average Rating	**5.839**	**2.926**	**5.390**	**6.000**	**5.268**	**6.292**	**7.048**	**6.707**	**6.975**	**6.536**

Group Average: 5.08292683

Group Average: 6.71219512

Appendix N

Examples of IL Introduced into Experimental User Guide

Control: Original Text in Control Version of User Guide
Experimental: Edited Version in Experimental Version of User Guide

1	Control - Page 4:	After calling a name or title using a *QuicKey* you MUST enter text before pressing enter.
	Control - Page 10:	Before pressing enter, you MUST enter some text once you have called a name or title using a QuicKey.
	Experimental – Pages 4 & 10:	If you use a QuicKey to insert a name or title, you MUST enter text before pressing enter.
2	Control - Page 4:	Pressing the enter key causes the text to be associated with the sound file.
	Control - Page 10:	When you press the enter key, the text is associated with the sound file.
	Experimental – Pages 4 & 10:	When you press the enter key, the text is associated with the sound file.
3	Control - Page 5:	Click *New* to create a new blank document.
	Control - Page 13:	Select the *New* icon to open a new blank document.
	Experimental – Pages 5 & 13:	To create a new blank document, click *New.*
4	Control - Page 6:	Click *Copy* to copy the selected text to the clipboard.
	Control - Page 14:	To copy the selected text to the clipboard, click *Copy.*
	Experimental – Pages 6 & 14:	To copy the selected text to the clipboard, click *Copy.*
5	Control - Page 6:	Click *Paste* to paste the contents of the clipboard to the log.
	Control - Page 14:	Select the *Paste* icon to paste the contents of the clipboard to the log.
	Experimental – Pages 6 & 14:	To paste the contents of the clipboard into the log, click Paste.
6	Control - Page 8:	From the *File* menu, click Format and select Paragraph.
	Control - Page 8:	Click *Format* on the *file* menu, and select *Paragraph.*
	Control3 - Page 8:	Go to the file menu and click *Format.* Now select *Paragraph.*
	Experimental – Page 8:	From the File menu, click Format and select Paragraph.
7	Control - Page 9:	QuicKey is a type of database that allows administrators to pre - program frequently used names, terms, portfolio titles, procedural statements etc. for quick retrieval and automatic insertion into the log text as described earlier on.
	Control - Page 17:	QuicKey is a database module that facilitates the programming of frequently used names, terms, portfolio titles, procedural statements etc. to allow the quick retrieval and automatic insertion into the log text.
	Experimental – Pages 9 & 17:	QuicKey is a type of database that allows administrators to pre - program frequently used names, terms, portfolio titles, procedural statements, for example, so that they can be retrieved quickly and inserted automatically into the log text.

8	Control - Page 9:	Rather than users typing in long names, terms or other phrases all that a user needs to do is enter a pre - defined shortcut.
	Control - Page 17:	Instead of typing in long names, terms or other phrases, all users need to do is enter a pre - defined shortcut.
	Experimental – Pages 9 & 17:	Instead of typing in long names, terms or other phrases, all all you need to do is enter a pre - defined shortcut.
9	Control - Page 9:	The QuicKey option is enabled when there is a check mark in front of the Use QuicKeys option.
	Control - Page 18:	If there is a checkmark in front of the *Use QuicKeys* option, the QuicKey option is enabled.
	Experimental – Pages 9 & 18:	The QuicKey option is enabled when there is a check mark in front of the Use QuicKeys option.
10	Control - Page 9:	If no checkmark appears in before the *Use QuicKeys* option, the QuicKey option is disabled.
	Control - Page 18:	If there is no checkmark before the *Use QuicKeys* option, the QuicKey option is disabled.
	Experimental – Pages 9 & 18:	The QuicKey option is disabled when there is no check mark in front of the Use QuicKeys option.
11	Control - Page 12:	The Toolbar is enabled when there is a check mark in front of the Toolbar option and is disabled when there is no check mark in front of the Toolbar option.
	Control - Page 15:	The *View Toolbar* option is enabled when there is a check mark to the left of the *View Toolbar* option.
	Experimental – Pages 12 & 15:	If there is a check mark beside View Toolbar, the toolbar is displayed.
12	Control - Page 12:	The Toolbar is enabled when there is a check mark in front of the Toolbar option and is disabled when there is no check mark in front of the Toolbar option.
	Control - Page 15:	To hide the toolbar, make sure there is no checkmark beside the Toolbar option.
	Experimental – Pages 12 & 15:	If there is no check mark beside View Toolbar, the toolbar is not displayed.
13	Control - Page 12:	The Status Bar is enabled when there is a check mark in front of the *Status Bar* option and is disabled when there is no check mark in front of the *Status Bar* option.
	Control - Page 15:	If there is a check mark to the left of the view Status Bar option, the status bar will be displayed.
	Experimental – Pages 12 & 15:	If there is a check mark beside View Status Bar, the status bar is displayed.
14	Control - Page 12:	The Status Bar is enabled when there is a check mark in front of the *Status Bar* option and is disabled when there is no check mark in front of the *Status Bar* option.
	Control - Page 15:	The *View Status Bar* option is disabled when there is no check mark to the left of the view Status Bar option.
	Experimental – Pages 12 & 15:	If there is no check mark beside View Status Bar, the status bar is not displayed.