A SEMIOTIC APPROACH TO THE USE OF METAPHOR IN HUMAN-COMPUTER INTERFACES

A thesis submitted for the degree of Doctor of Philosophy

by

C. Condon

Department of Information Systems and Computing, Brunel University

October 1999

Abstract

Although metaphors are common in computing, particularly in human-computer interfaces, opinion is divided on their usefulness to users and little evidence is available to help the designer in choosing or implementing them. Effective use of metaphors depends on understanding their role in the computer interface, which in turn means building a model of the metaphor process. This thesis examines some of the approaches which might be taken in constructing such a model before choosing one and testing its applicability to interface design.

Earlier research into interface metaphors used experimental psychology techniques, which proved useful in showing the benefits or drawbacks of specific metaphors, but did not give a general model of the metaphor process. A cognitive approach based on mental models has proved more successful in offering an overall model of the process, although this thesis questions whether the researchers tested it adequately. Other approaches which have examined the metaphor process (though not in the context of human-computer interaction) have come from linguistic fields, most notably semiotics, which extends linguistics to non-verbal communication and thus could cover graphical user interfaces (GUIs).

The main work described in this thesis was the construction of a semiotic model of human-computer interaction. The basic principle of this is that even the simplest element of the user interface will signify many simultaneous meanings to the user. Before building the model, a set of assertions and questions was developed to check the validity of the principles on which the model was based. Each of these was then tested by a technique appropriate to the type of issue raised. Rhetorical analysis was used to establish that metaphor is commonplace in command-line languages, in addition to its more obvious use in GUIs. A simple semiotic analysis, or deconstruction, of the Macintosh user interface was then used to establish the validity of viewing user interfaces as semiotic systems. Finally, an experiment was carried out to test a mental model approach proposed by previous researchers. By extending their original experiment to more realistically complex interfaces and tasks and using a more typical user population, it was shown that users do not always develop mental models of the type proposed in the original research. The experiment also provided evidence to support the existence of multiple layers of signification.

Based on the results of the preliminary studies, a simple means of testing the semiotic model's relevance to interface design was developed, using an interview technique. The proposed interview technique was then used to question two groups of users about a simple interface element. Two independent researchers then carried out a content analysis of the responses. The mean number of significations in each interview, as categorised by the researchers, was 15. The levels of signification were rapidly revealed, with the mean time for each interview being under two minutes, providing effective evidence that interfaces signify many meanings to users, a substantial number of which are easily retrievable.

It is proposed that the interview technique could provide a practical and valuable tool for systems analysis and interface designers. Finally, areas for further research are proposed, in particular to ascertain how the model and the interview technique could be integrated with other design methods.

Contents

List of figures

List of tables

Acknowledgements

Some of the earlier work in this thesis was carried out in collaboration with fellow researchers from RACE Project R2094: MITS (Metaphors for Integrated Telecommunications Services). Where specific work was undertaken by other members of the team this has been acknowledged within the main text. It is often difficult to remember exactly where an idea originated and I would therefore like to offer my thanks to all other members of the project for the work they did and my apologies to them if I have inadvertently overlooked any specific acknowledgements which they deserve. In particular, some of the initial literature searches of metaphor theory and rhetoric which contributed to the introductory material in sections 2.1.1, 3.1.1 and 3.3.1 were originally carried out by Roger Knott of Loughborough University of Technology.

Although all other work in the thesis is entirely my own responsibility, I would have been unable to complete it without the help of my supervisors. I would like to thank Professor Alty at Loughborough where I originally registered, and Professor O'Keefe at Brunel where I completed it. I would also like to thank Hans Panse and Stephan Keuneke of BIBA for their assistance in setting up and running the experiment described in sections 4.4 - 4.8. Finally, I would like to thank Mark Perry from Brunel and my daughter Louisa Condon from the University of Sussex for carrying out the content analysis described in section 5.4.

1 Introduction

1.1 Subject area

This work is concerned with the use of metaphors in the human-computer interface and their role in the relationship between the user and the computer, and thus falls into the area known as human-computer interaction (HCI). The target area in which it is hoped the results will prove useful is that of systems analysis and design, with particular regard to the specification and design of the interface.

Some of the earlier work described in this thesis was carried out as part of the MITS project (Metaphors for Integrated Telecommunications Services), funded by the European Commission as part of RACE (Research in Advanced Communications for Europe). Although that project studied problems associated with multimedia communications and computer supported cooperative work (CSCW), this restriction does not apply to this thesis.

1.2 Research problem

Many authors have claimed that interface metaphors can be useful to the user, particularly for new users learning how to use a system. Others have identified problems caused when metaphors are taken too literally by users. The preliminary research in this thesis shows that metaphors are common, even in 'metaphor-free' interfaces. Other authors have shown that, even where no explicit metaphor has been used in the interface, problems can be caused by metaphors that have been introduced by users. Thus the problem is not whether to use metaphors in the interface but how to introduce metaphors and which metaphors to use.

Designers who have introduced metaphors with the intention of assisting users have done so on an ad hoc basis. A common assumption is that metaphors should be based on familiar aspects of the users' workplace, such as the materials and tasks found in an office environment. Although this appears to make sense, introducing the unfamiliar in terms of the familiar, there appears to be very little empirical research to test the approach. Much of the argument about the use of interface metaphors, particularly from those opposed to the use of metaphor, has concentrated on graphical user interfaces (GUIs). However, it is often possible to introduce the same metaphor as an icon in one interface and as a word in another. Whether the effectiveness of the metaphor depends on the type of interface remains unanswered.

Ideally, the interface designer needs guidelines on when to use metaphors, which metaphors to use, and how to portray them. Metaphors are rarely introduced in isolation and the designer must also understand how they should be mapped to the functionality of the system and whether they can be mixed with other interface metaphors. Some HCI researchers have offered principles to guide some of these decisions, whilst systems manufacturers often offer style guidelines to ensure consistency across applications. Current guidelines are, however, based on personal experience, common sense and aesthetics. More effective guidance depends on a better understanding of the process of metaphor. It is not possible to directly observe the mental processes through which a user understands and uses an interface metaphor, but it should be possible to build a model of this interaction. A number of disciplines offer possible approaches which could be used to create such a model.

1.3 Possible approaches

1.3.1 Standard HCI methods

One possible approach might be to look at, and possibly extend, existing HCI methods in software ergonomics and interface design. Many authors offer useful methods for building interfaces and for evaluating their usability, but these are independent of any underlying metaphor. It would certainly be possible to evaluate interfaces based on different metaphors and find which is the most usable. Unfortunately, such data does not show *why* a particular interface is better. Thus the particular implementation might be more important than the metaphor chosen; it is certainly possible to build bad interfaces with the best of metaphors.

Where metaphor is discussed in the HCI literature, it is generally justified as presenting novel functions in terms of well-understood ones. This is usually interpreted as using metaphors drawn from the environment in which the interface is to be used. However, very little evidence has so far been presented to support this assertion.

1.3.2 Computer science

The three most important elements involved in this research are the computer, the human and the metaphor. One approach might be to begin with studies of the computer and extend these. Computing and metaphor have always been fundamentally linked. Even the central principles of computing are close to those of metaphor: programming one machine to act *as if* it is another. Although this indicates how computers might support metaphor, this approach does nothing to indicate which metaphors are likely to be most helpful to users.

Computer science certainly employs many metaphors, such as 'objects' but treats them as mathematical constructs. Mathematics is itself largely based on metaphors but having established the original metaphor, mathematics employs abstraction to form a self-contained system. Thus, for example, the mathematical concept of a 'set' comes from our common usage of the word in concepts such as a tea-set. While the metaphor may have provided the original inspiration, it is immediately abandoned and replaced by an abstract mathematical definition of the concept. This route is uni-directional: the properties of a mathematical set cannot be used to assess the usability of a tea-set.

Computer science forms a similar type of abstraction which might be useful in creating a formal specification of a metaphor. However, abstraction and formalisation can only occur after the characteristics of a metaphor and the process through which it works have been understood. Computer science offers no techniques for building this initial understanding.

1.3.3 Cognitive psychology

The second element of this study is the human: metaphor is a cognitive process. If we look at the way that metaphors appear to work in the human mind, we find this is easily expressible in terms of mental models. This approach has been widely used in HCI, comparing the designer's model of the system with the user's model and comparing both to the physical model. It is usually asserted that mis-matches between the three models can create problems for the user. It is relatively easy to extend this approach to the role of metaphor in the interface by including the user's mental model of the domain from which the metaphor is drawn.

When a new concept or function is to be introduced it can be attached to this existing mental model using the process of metaphor. Over time, a new model will be created, detaching itself from the original and forming a new, separate definition of the new concept. At least one experiment has been based on this model of metaphor. The experiment was based on the idea that we form models of the systems we use and that the accuracy of these models affects how easily we can use the systems. This approach is further explored in this thesis.

1.3.4 Linguistic approaches

The third element to be considered is the metaphor itself. Metaphor has been widely studied in literature and has been examined in a number of areas of linguistics. The question arises as to whether it is valid to apply studies based on language to computer interfaces. Many interface metaphors are presented as images or actions rather than words and it needs to be established whether this affects the nature of the metaphor. A second issue is whether the elements of a computer interface have equivalent syntactical and semantic power to a natural language, or at least sufficient power to support true metaphors.

Although metaphor was originally studied in the context of rhetoric, literature and poetry, more recent approaches have extended this work to other fields such as advertising and film. If these studies are valid then it is obviously possible to extend this approach to computer interfaces. This type of approach has not been widely employed in HCI and appears to show great potential for the study of metaphor and, perhaps, other aspects of human-computer interaction. It therefore forms the main field of study in this thesis.

1.4 Research objectives

As explained when considering the research problem above, the most important concern is that there is currently no effective model of the metaphor process. Both cognitive psychology and linguistic approaches are assessed in terms of their potential for building such a model. Either approach depends on a number of assumptions which must be tested before carrying out any study of the full model.

Two approaches could be made when assessing the usefulness of the model. The first is its potential use by future HCI theorists and researchers, which would depend on its internal consistency and whether the model adds to our understanding of human-computer interaction. The second approach is to consider its potential use by interface designers. The research problem outlined above is a practical problem faced by many designers who need guidance. This thesis concentrates on the second of these approaches, looking for practical justification of the model in terms of its usefulness to designers, although it is acknowledged that this will also depend to some degree on establishing a solid underlying theoretic basis.

1.5 Research methods

A number of preliminary studies must be made before a model can be developed. Short studies are used to assess the potential approaches introduced above. In addition to these, a suitable linguistic or grammatical method must be used to test the assumption that the computer interface can be considered in a comparable manner to a linguistic system. Following the results of the preliminary studies, the most promising approach is used to develop a model of the metaphor process.

It is impossible to test whether the model is 'right' in the sense of accurately representing the thought processes of the user, in that many relevant thought processes are not conscious. Other ways of testing the model must therefore be considered. The model could be tested as a predictor of user behaviour but this only provides evidence for its descriptive power. Another approach is to look at the particular model, examining its implications for the analysis and design process and testing those implications. As outlined above, it is the second of these approaches that is followed.

A number of research methods are considered and the most suitable chosen. This is a simple interview technique to be carried out in the workplace with users of two types of application. If successful, this can not only provide evidence for the concordance of the model with the users' understanding but also provide a demonstration that the model could be used in similar circumstances by interface designers to provide practical assistance in the design of the interface.

1.6 Dissertation outline

The contents of this dissertation are given in approximately chronological order. Chapter 2 looks at the background to the work, considering previous research into computing metaphors and the potential approaches which could be used. Previous work, particularly that of Carroll, is then examined, showing some of the problems and benefits of metaphor in computing. The chapter also examines studies of metaphor by Eco and others, including its role in the development of languages. Lakoff's thesis that metaphor plays a major role in human cognition is then considered. Finally, the chapter looks at some of the most important examples of metaphor in computing and communications.

Chapter 3 then lays out a provisional model for human-computer interaction, together with proposals for the useful application of this model to interface design. A superficially similar approach is considered and shown to be complementary rather than a direct equivalent. Finally, the chapter lays out the assumptions made in the development of the model and some questions which it raises, together with proposals for testing them. Chapter 4 describes the preliminary research used to test these underlying assertions and questions. The short experiments and studies use a variety of techniques from different fields as appropriate to the assertion or question under consideration.

Potential research methods are considered in chapter 5 and the particular study method developed and described. This is the interview based technique referred to above. A series of case studies are then described in chapter 6 in which the technique is used with two different user groups. The results of the interviews demonstrate the potential usefulness of the approach. Finally, chapter 7 draws together results and conclusions from the previous tests and experiments. Suggestions are also made for further research in testing the model and applying it to other areas of human-computer interaction.

2 Computing and metaphor

2.1 Metaphor

2.1.1 Introduction

The word 'metaphor, comes from the Greek μεταπηορα (metaphora), which literally means to 'transfer' or 'convey'. This literal meaning remains in modern Greek but even in ancient Greece, 'metaphor' had a second meaning: "Metaphor is the transport to one thing of a name which designates another" (Aristotle, *Poetics*). It is this meaning, itself originally metaphorical, which has been adopted by English and other European languages.

Metaphor is an example of a trope, the rather inadequate dictionary definition of which is a 'figure of speech'. The next chapter will look at tropes more closely. However, most linguists agree that tropes are more than parts of speech, following Richards' definition:

> The Traditional theory ...made metaphor seem to be a verbal matter, a shifting and displacement of words, whereas fundamentally it is a borrowing between and intercourse of *thoughts*, a transaction between contexts. *Thought* is metaphoric, and proceeds by comparison, and the metaphors of language derive therefrom. (Richards 1936, p.96)

Richards also introduced what is now a standard terminology for the components of a metaphor:

- **Tenor**: the original concept
- **Vehicle**: the second concept 'transported' to modify or transform the tenor
- **Ground**: the set of features common to the tenor and the vehicle
- **Tension**: the effort demanded to span the gap between the tenor and the vehicle

Confirmation that metaphor deals with thoughts rather than simply words comes from its role in the development of new concepts in science. (Leatherdale 1974). Eileen Cornell Way gives some examples of the importance of metaphor to science:

> The use of metaphor to extend our concepts in science is legendary: the Bohr model of the atom uses the structure of the solar system, Maxwell's represents an electrical field in terms of the properties of a fluid, atoms as billiard balls, etc. Thus, even science is not the paradigm of literal language it was once considered to be; rather metaphor is vital to the modelling processes that result in advances in science. (Cornell Way 1991, p.8)

The power of metaphor in science is not always beneficial. For example, Huygens view of light as continuous waves, which he likened to those caused by a stone dropping into water (Eisenberg 1992, p.144), led to confusion and argument when other experiments appeared to show light acting in a particulate manner (Feynman 1990, p.15). It appears that the metaphors that help early understanding can sometimes hinder the further development of that understanding.

2.1.2 Metaphor in computing

Although metaphor is a thought-process that can be expressed through language, is it equally valid to express metaphor through computation? Certainly the correspondence between computation and language at a low level is very strong:

> To each type of language there corresponds an appropriate class of abstract machines which recognize precisely languages of that type. In the general case, the abstract machine appropriate for the type-0 grammars is the Turing machine, a fact that restates Turing's thesis in the form of Chomsky's form of Post canonical systems.
> (Brady 1977, p.88).

Following this definition, Brady also provides a mathematical proof (Brady 1977, p.88-90). Alternative proofs may be found in Cooke & Bez

(1984, p.265-73) or many other Computer Science textbooks. These demonstrate that each of Chomsky's grammars generate a language that is computable by a specific class of abstract machine, from Finite State Machines upwards. The highest level of language, generated by a Chomsky type-0 grammar, is computable by a Turing Machine (TM) and also approximately corresponds to the class of natural human languages.

Both Chomsky's grammars and TMs are idealised forms. In particular, a TM demands unbounded, though not infinite, storage capacity and unlimited time. In other words, it is not always possible to predict in advance how much storage a program will require, or how long it will run, except by running it on a TM. By contrast, computers have limited storage capacity and typically include control programs that will halt a program that recurses more than a certain number of times. These limitations can be seen as directly comparable to the limits of human memory and our inability to handle more than a limited amount of clause-nesting in a sentence. In practice, computer languages are comparable to natural languages, as we normally use them.

Computer programming is also, generally, a verbal activity in that it is based on words with a linear syntax, its semantics depending on the ordering of these words. In this it is close to most human languages, although a natural language does not have to be verbal or have a linear syntax. For example, natural sign languages have a spatial syntax quite different from the linear syntax of verbal language (Sachs 1989, p.76).

Thus, there is a mathematical concordance between computers and language at a basic level. The comparable syntactic structures imply that similar semantic structures *could* exist on top of these. It does not confirm that they do.

One reason to suggest that computing will involve metaphor is that the basic principle of programmable machines is very close to that underlying metaphor. Even before the existence of computers, Turing showed that a Universal Turing Machine (UTM) could be programmed to behave exactly as any other Turing Machine. In other words, one abstract machine may be 'transported' to another in a similar manner to the principle of metaphor 'transporting' a concept from one context to another. Alan Kay goes further:

> The protean nature of the computer is such that it can act like a machine or like a language to be shaped and exploited. It is a medium that can dynamically simulate the details of any other medium, including media that cannot exist physically. It is not a tool, although it can act like many tools. It is the first metamedium, and as such it has degrees of freedom for representation and expression never before encountered and as yet barely investigated. (Kay, quoted in Laurel 1993, p.32)

A simple practical example of this comes when one type of computer is used to emulate a different computer, forming a virtual machine. Similarly, software can be used to create virtual input and output devices or virtual discs. Conversely, this power means that computing and information systems can themselves become powerful metaphor vehicles in areas such as management (Jackson 1995).

Machine code and assembly language keep to the step-by-step instructions of the Turing Machine but higher level languages can involve the introduction of structures, such as 'objects'. These are unarguably metaphors, introduced to assist the programmer, and have no existence in the low level machine code which is generated. Programming is a very specialised form of human-computer interaction and most computer users do not use full programming languages, but metaphor is also prominent in the interfaces used by ordinary computer users, particularly graphical user interfaces (GUIs).

It might be argued that, in contrast to computer languages, a GUI is not a full language. At the extreme, consider the very limiting definition of language given by Weinrich:

> All languages are information-conveying mechanisms of a particular kind, different from other semiotic mechanisms which are not language. Thus we could rule out, as non-language, systems whose sign vehicles are not composed of discrete recurring units (phonemes); systems which have unrestricted combinability of signs (i.e. no

> grammar); systems whose signs are iconic; perhaps even such systems - to add a pragmatic criterion - as are not used for interpersonal communication. (Weinrich 1966, p.142).

According to this definition, non-programmable GUIs are definitely not true languages nor, even, are computer languages. However, they do classify as what Weinrich refers to above as 'semiotic mechanisms' or 'systems [of] sign vehicles'. Semiotics may be seen as a superset of linguistics, dealing with all semiotic systems: both language and other 'systems of sign vehicles'. The next chapter will look further at semiotics; for now it is sufficient to point out that semioticians regard any semiotic system as capable of carrying a metaphor.

2.1.3 Metaphor and other tropes

Metaphor is an example of a trope: a non-literal method of description. Appendix A lists the common tropes, together with other potentially relevant rhetorical devices. Other tropes work in a similar manner to metaphor, most notably simile and analogy. The difference between simile and metaphor is whether the drawing together of the two concepts is implicit or explicit, respectively:

Metaphor: "The Macintosh interface is a desktop"

Simile: "The Macintosh interface is *like* a desktop"

Analogy differs from these in drawing parallels between extended processes or narratives. For example, the political endeavours of John Iselin's family in Richard Condon's book, 'The Manchurian Candidate' (Condon 1973) are often described as analogous to those of the Kennedys and were certainly intended to be so by the author. However, many people also saw a post hoc analogy in the death of John Iselin in the book, though it was written before John or Robert Kennedy was killed. Analogy and metaphor, unlike simile, do not have to be intended by their creators.

It might be argued that, for example, the Macintosh's desktop metaphor's explicit nature makes it a simile. Others might argue that its extension to so many sub-metaphors, such as folders and documents, means it is more truly an analogy. However, the underlying thought processes, that of transporting a concept from one context to another, is the same in all these cases. Analysts such as Lodge and Jakobson have identified metaphor as one of two master tropes, with analogy and simile as sub-classes of metaphor (Jakobson 1956; Lodge 1990). The second master trope is metonymy, with synecdoche as a sub-class, in which a part or an attribute stands for the whole, or the whole stands for a part. For example, we might say, "Netscape announced a new attack on Microsoft today", rather than "A spokesperson from Netscape...."; or "Sculley produced too many low-end machines", rather than "Apple factories, when the company was headed by Sculley, produced...". Lodge sees the two master tropes as central to discourse:

> Metaphor is derived from similarity: metonymy and synecdoche from contiguity. As soon as discourse deviates from strictly literal, denotative reference, it will tend to do so either in the form of metaphor and simile, or in the form of metonymy and synecdoche.
> (Lodge 1990, p.151).

Eco (1985, p.251) agrees with the primacy of "metaphoric mechanisms and metonymic mechanisms; to these one can probably ascribe the entire range of tropes, figures of speech, and figures of thought." There is also some argument about the relationship between metonymy and metaphor. Jakobson (1956) sees metonymy as a different principle of organisation to that of metaphor, as do Wellek and Warren (Wellek 1976). Lodge (1977, p.79-80) repeats Jakobson's argument, using Jakobson's evidence of different forms of aphasia (Jakobson 1956, p.58). In one form, patients have problems in talking of anything not present, generating apparent metaphors; in the second form, patients make 'metonymic' mistakes such as substituting 'knife' for 'fork' or 'smoke' for 'pipe'. However, neither Jakobson nor Lodge presents any evidence that these patients are using the same processes of metaphor and metonymy as those used in normal speech or literature. Indeed, Lodge has some difficulty with the extensive use of metonymy in literature, which he sees as essentially metaphoric. He explains this by saying that even though metonymy is used, "The literary text is always metaphoric in the sense that when we interpret it...we make it into a total metaphor: the text is the vehicle, the world is its tenor." (Lodge 1977, p.109). This does not appear to fully resolve the contradiction.

Others see one trope as a form of the other: Whittock (1990) sees metonymy as a type of self-referential metaphor, whereas Eco (1985) argues that metaphor depends on metonymy in that it abstracts a feature or features which the two domains have in common, i.e. the ground. Whichever of these arguments is accepted, it appears likely that the role of metaphor in user interfaces cannot be examined without some attention to the parallel role of metonymy.

The primacy of the two master tropes is widely recognised as being not only central to discourse but also to the development of language, particularly in the case of metaphor:

> The majority of our messages, in everyday life or in academic philosophy, are lined with metaphors. The problem of the creativity of language emerges, not only in the privileged domain of poetic discourse, but each time that language – in order to designate something that culture has not yet assimilated.... must *invent* combinatory possibilities or semantic couplings not anticipated by the code. (Eco 1985, p.262).

Some see the two processes as central to more than just language. As Lodge explains, "Metaphor and metonymy are in fact manipulations of two processes that are basic to language, and perhaps to all perception and representation – selection and combination." (Lodge 1990, p.150). Lakoff and Johnson take, if anything, a stronger view:

> [Most] people think they can get along perfectly well without metaphor. We have found, on the contrary, that metaphor is pervasive in everyday life, not just in language but in thought and action. Our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature. (Lakoff 1980, p.3).

There is also a third, minor trope separate from metaphor and metonymy – irony. Irony is rare in human-computer interface design, although the titles of 'yacc' (yet another compiler compiler) or 'Yahoo!' (Yet another hierarchically organised object) could be seen as examples, as could the use of the 'Jack-in-a-Box' icon for the far from playful ResEdit program on the Macintosh.

Language evolves and develops new terms through metaphor and metonymy, as the references from Lodge and Eco make clear. This is also why these two tropes are so important to computing. Less than fifty years ago there were no computer languages; today there are probably many hundreds, if not thousands, of computer languages and dialects, in addition to many different user interfaces. Both programming languages and user interfaces require descriptive systems for the new concepts computers introduced, such as programs, files and disc drives. As with other semiotic systems, it is to be expected that the two main routes to describing these new concepts will be metaphor and metonymy.

Although this thesis is more concerned with metaphor, metonymy is also used in the development of computing terms. Unfortunately, the features which initially distinguish a concept or object may not be the most useful in the long term. For example, when floppy disc drives were first used in personal computers, the term 'disc' was a useful distinction from the cassette tape then in use, echoing the use of the same terms for music cassettes and 12" discs; the term 'floppy' helped distinguish the new type of disc from the removable 'hard' discs then used on mainframe computers. Now, $3^1/_2$" discs have both their flexibility and disc-shape hidden in a hard, square case.

2.1.4 Dead metaphors

The processes of metaphor and metonymy are usually seen as the first stage in the formation of a new term. After a time, the origins of a word or expression tend to be forgotten and it becomes an accepted part of the language, a process known as assimilation. In the case of a metaphor, we describe the metaphor as 'dead'. The death of a metaphor depends on many factors and is personal; computer metaphors that have died for those working in the field will be seen afresh as metaphors by new users.

Simply because a metaphor has been in existence for a long time does not necessarily mean that it is dead. Lakoff and Johnson argue that certain types of metaphor are fundamental to all language, built up from our common experiences when growing up. One such concept they identify is that of the 'orientational metaphors' (Lakoff 1980, p.14-24). For example, the use of the word 'right' in contrast to 'wrong' probably has metaphoric roots in that right-handed people are seen as doing things the 'right' way. In contrast, left-handed people do things in a manner that is unnatural or 'sinister' – the Latin word for 'left'. This metaphor is probably now dead for most people. For example, a user confronted with the following dialogue would be unlikely to experience confusion or discomfort, apart from its departure from the Apple Human Interface Guidelines:

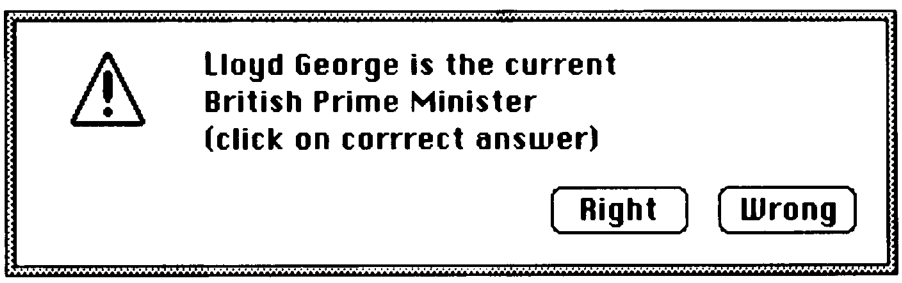


Figure 2.1: 'Right' as a response in left position.

We usually talk of 'right and wrong' and the dialogue follows our normal verbal ordering, placing the more important term first: it does not matter that the word 'Right' is on the left. However, the following dialogue would also be acceptable:

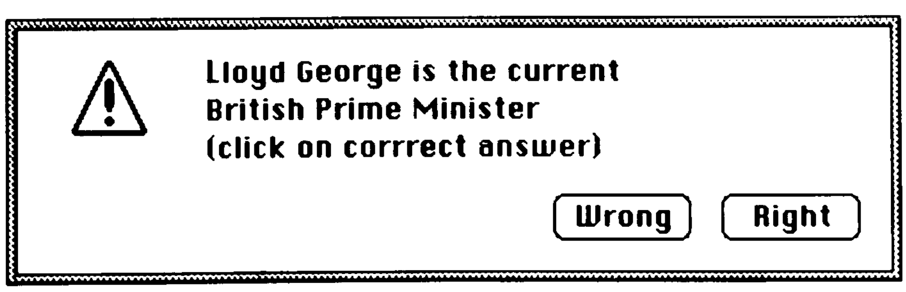


Figure 2.2: 'Right' as a response in right position.

To treat a dead metaphor as being alive is not likely to damage the user's understanding, but the dangers of ignoring metaphors are much greater, and metaphors can persist for a long time without completely dying. For example, we talk of 'high' temperatures or 'depths' of cold. Consider the fictional graph below, in which hotter temperatures are shown as physically lower:

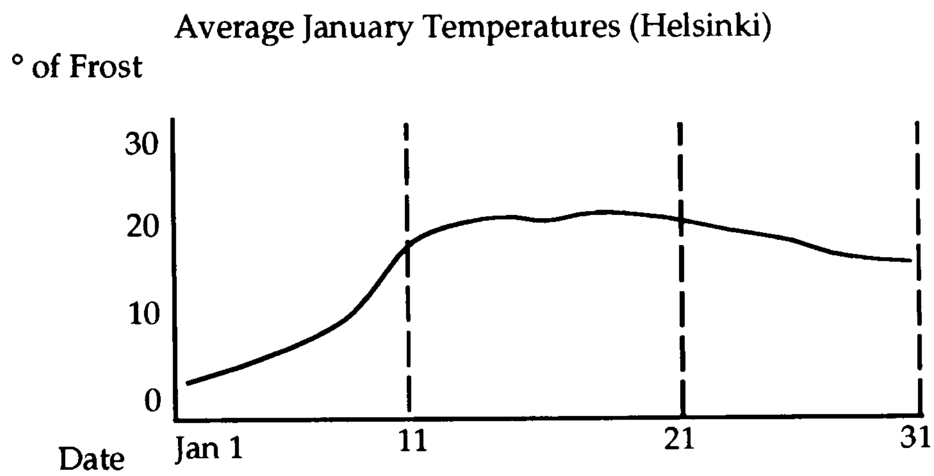


Figure 2.3: Inverted temperature scale.

It is not just the association with the numbering on the temperature scales that dictates this relationship. The graph above uses degrees of frost as the scale, in which a larger number refers to greater cold. People to whom I have informally shown this graph find it difficult to see it in this way and persist in seeing 'hotter' as being 'higher'. Though people may no longer be conscious of the metaphor, it has not completely died.

As a simple guide, a metaphor can be considered dead once the metaphoric meaning is listed in the dictionary as a meaning in its own right. Goatly (1997, p31-35) discusses why some metaphors die but not others, and shows that the situation is more complex, giving examples of metaphors which he sees as 'dead', 'dead and buried', 'sleeping' and 'tired'. "Dictionaries are certainly the cemeteries and the mortuaries, definitely the dormitories, and generally the resting place for the populations of metaphors." (Goatly 1997, 31).

2.2 Other ways of naming new concepts

Languages also evolve in other ways that need to be distinguished from the two main tropes. Although this thesis is concerned with metaphor and, to a lesser extent, metonymy, it is important to be aware of these mechanisms to avoid confusing them with metaphor or metonymy.

2.2.1 Word importation

A common way of naming new concepts is through word importation. A language imports words from other languages as the associated culture imports new objects or concepts, such as 'sauna' or 'Zeitgeist'. At first glance, it might be thought that the adoption of English words in a computer language or interface is an example of importation. Take, for example, the use of the term 'bookmark' to apply to a marker for a frequently accessed page in a Web browser.

It might appear that the term has been imported from English to the interface language. However, the term 'bookmark' was already familiar to the designers and users of the application and was applied to a new, unfamiliar concept, i.e. the process of metaphor. Word importation takes place when new, unfamiliar words are introduced together with new, unfamiliar concepts. Thus, for example, we have recently imported both the word and the concept of 'ombudsman' from Sweden. This was a new concept described by a new word, not a new concept named with a familiar word as was the case with 'bookmark'. In German, the term 'bookmark' has been imported and is seen as a new word (in German) for a new, computer interface concept.

2.2.2 Neologisms

A few methods of creating entirely new words, or neologisms, can be used. Acronyms can be formed, RAM (Random Access Memory) and ROM (Read Only Memory) being examples. Alternatively, new words can be generated from historical languages, such as Norbert Wiener's coining of 'cybernetics' – "We have decided to call the entire field of control and communication theory... by the name *Cybernetics* which we form from the Greek $\chi\upsilon\beta\varepsilon\rho\nu\eta\tau\eta\zeta$ or *steersman*." (Wiener 1961, p.11). In this, as in many examples, metaphor is involved, but most people would not have sufficient knowledge of Greek to see this, seeing it simply as a new word. Other new terms can be taken from peoples' names, such as the Bernoulli drive or the Ada language.

2.2.3 Onomatopoeia

Onomatopoeia, in which a word imitates the sound made by the object, is rarely appropriate to computing. Some computer concepts are described by onomatopoeic words, such as 'beep' and 'click', but these words were already current in the language when they were adopted as computing terms.

The principle of onomatopoeia can be extended to words that sound like other words. For example, many words beginning 'sl' have slippery or icy meanings (slip, slime, slide, slant, slope, slalom, sled, sleigh, sledge, sludge, sleet, slop) or are associated with sleeping (sleep, slumber), both obviously onomatopoeic. This has been extended to metaphorically 'slur' people or places (sleazy, sloppy, slippery, slovenly, slouching, slob, slattern, slut, slag, slug, slum). Thus, when inventing a new term, an 'sl' word will automatically have such connotations, such as Lewis Carroll's description of the 'slithy toves' in Jabberwocky (Carroll 1982, p.134). There are a few examples where this process may have taken place in the choice of computing metaphors. As explained below, the term 'bug', originally came from military slang. However, the choice of this particular term, rather than 'insect', 'gremlin' or 'spanner in the works', may well owe something to its similarity to other, less polite 'b-words' which could come to mind when coming across such a fault.

2.2.4 Back-formation

Languages also develop through back-formation, where a new term is formed from another, even though an existing form exists. For example, computer users commonly use the term 'to input' data, a back-formation from the noun 'input', rather than the original verbal form 'to put in'. The two forms have now diverged and 'to input' is almost always restricted to its meaning in computing.

2.2.5 Slang and jargon

Natural languages include many words and expressions which form communal codes, specific to social groups of one sort or another. Terms may be associated with a geographical area (dialect), a profession (register, jargon), or a community such as prisoners (cant, argot), while other words are in general use but not accepted in 'polite society', such as swearing (Andersson 1992, p.67-90). Some terms remain outside mainstream usage for centuries, while others rapidly move from one social group to another, and even into the mainstream 'polite' language. Indeed, the term 'slang' was itself once slang (Andersson 1992, p77).

Terms enter the computing and communications community from other social groups. Some have argued that computing is dominated by military slang, such as 'bug' and 'fragging' (Levidov 1989). Others terms come from science fiction and comedy: 'tekkie' (a technical person), formed by analogy to 'Trekkie' - an insulting term for a 'Trekker' or Star Trek fan; 'spamming' (flooding newsgroups with irrelevant, often repeated information), attributed by Crystal (1998, p.108) to the Monty Python 'Spam' sketch.

2.3 Interface metaphors

2.3.1 The desktop metaphor

Most conscious use of interface metaphors has been applied to graphical user interfaces (GUIs). Interface metaphors were undoubtedly used in non-graphical interfaces, as the discussion of the work of Carroll et al below explains, but there is little evidence that this stemmed from a conscious decision by the interface designers.

The best documented examples of the explicit use of metaphor for GUIs are the Xerox Star, the Apple Lisa and the Apple Macintosh. What is now usually known as the desktop metaphor began with the Xerox Alto and was refined on the Xerox Star. The designers chose what they referred to as the "physical-office metaphor" because the Star was intended as an office information system, so reflecting the familiar world of the potential users (Smith 1982, p.246). The designers saw their metaphor as providing a 'physical' environment rather than a language of interaction:

> The Desktop is the principle Star technique for realizing the physical-office metaphor. The icons on it are visible, concrete embodiments of the corresponding physical objects. Star users are encouraged to think of the objects on the Desktop in physical terms (Smith 1982, p.247)

The concept was further developed in the design of the Apple Lisa and Apple Macintosh, where it is usually known as the 'desktop metaphor' rather than the 'physical office metaphor'. Apple also provided design guidelines to ensure the consistency of the interface across third party applications (Apple 1987). The desktop metaphor has proved very successful, being adapted to front-end DOS with MS-Windows and UNIX with Open Look and Motif, but there is evidence that it is reaching the limits of its usefulness as new applications for computing arrive. Even when the Star was designed the designers were aware that "It is probably not possible to represent everything in terms of a single model" (Smith 1982, p.247) and were forced to look for a different metaphor for the records processing facility.

The most interesting of the incarnations of the desktop metaphor in the context of this thesis was its implementation on the Apple Lisa. Apart from the use of formal grammars in the analysis of programming languages, this was probably the first time that a semiotic approach to the design of the human-computer interface was used in the construction of a graphical interface. Nadin (1988) shows the way in which the Lisa interface was given a clear semantics as well as a regular post-fix syntax. For example, one semantic convention was that icons should represent nouns and menu-items represent verbs.

Nadin's work on the Lisa interface seems to have been largely neglected since. An examination of the Lisa's successor, the Macintosh, shows a confusion of prefix ('Duplicate') and postfix ('Find') commands. The Macintosh has also dropped the Lisa's convention that menu-items should all be verbs, mixing verbs ('Find') with nouns ('Information'), adverbial phrases ('By Size') and even a menu of colours.

Nadin referred to his approach as 'semiotic' rather than 'linguistic'. Linguistics deals only with languages, particularly those based on words, whereas semiotics looks at all the ways in which any symbol or sign can carry meaning to a person. As such, it appears to have the potential to help in the design of both text-based and graphical user interfaces. I will explore this potential further in the next chapter.

2.3.2 New interface metaphors

The suitability of the desktop metaphor is now being challenged by new areas of computing. These include CSCW (Computer Supported Cooperative Work), hypertext systems, multimedia, the Internet, VR (Virtual Reality) and portable computing. Physical desks are generally used for office tasks by individuals and it is difficult to extend this metaphor to cover group working or other types of task, such as editing of video or audio.

Various metaphors have been proposed for CSCW, the most common probably being the room. A number of researchers have independently explored the use of the room metaphor. Xerox PARC (Henderson 1986) developed a room concept to be used by one user at a time on a single machine, while the concept was extended to multi-user groupware by Bellcore (Root 1988) and in my own work (Condon 1990). We also explored the combination of the room metaphor for informal, real-time work with a form-based metaphor for formal, non-real-time work (Hämmäinen 1991). Other have expanded beyond the immediate room to include balconies, doors and corridors (Pemberton 1993).

As its name suggests, hypertext was initially developed from a book or document metaphor, with links taking the user from one page to another. Apple recognised the limitations of the desktop metaphor when dealing with hypertext and used a 'card index' metaphor for HyperCard. This conflicted with their existing interface guidelines based on the desktop (Apple 1987), to which they responded by providing a new set of guidelines for HyperCard (Apple 1989). The hypertext principle has now expanded to hypermedia and hyperspace. Some have expanded the book metaphor to cope with these more expansive demands (Rauch 1997) or extended it to libraries (Pejtersen 1988), while others have looked towards various extended physical spaces or communities, listed below in the context of the Internet.

Multimedia presents obvious problems, in that media such as sound and video are not usually handled in an office context. Apple effectively abandoned the desktop in finding suitable ways of presenting the QuickTime multimedia facilities, adapting the standard control panel of a video recorder (Apple 1991). The wisdom of this is, perhaps, arguable, in that the poor usability of video recorders has also been condemned (Thimbleby 1991). Various multimedia preparation programs have used other metaphors drawn from the existing media industry, such as films (the storyboard in Macromind Director), newspapers/magazines (page layout programs), studio equipment (mixers for sound manipulation programs).

The combination of hypertext and multimedia on the Internet has led to a series of communications or link-based metaphors, such as the World Wide Web, the Information Superhighway, or simply, the Net. Many of the suggested interfaces for future systems are based on VR and a number of metaphors have been suggested for managing these virtual spaces. Many are based on extended spaces and landscapes or on various types of community. These include fields, villages, rivers and highways (Florin 1990), farms, including information fields and swamps (Bernstein 1993), information forests (Rifas 1994), or urban metaphors such as the city (Dieberger 1994b; GeoCities 1998). Others have even suggested non-human communities such as the ant colony (Bilchev 1993).

A separate, very popular category of metaphor is that of the interface agent or guide to show the user around (Oren 1990; Laurel 1990; Isbister

1995; Rich 1996; Lieberman 1997). This concept has now been widely adopted commercially, particularly for help facilities, in applications such as WordPerfect 6.1 (the Coach) and Office '97 (the Office Assistant).

It is finally worth noting that the size of the interface also affects which metaphors might be suitable. There is a tendency to choose metaphors for smaller devices which correspond to their size. Portable computers are often based on a book metaphor, such as Alan Kay's Dynabook (Kay 1990) or the Apple PowerBook. Palm-held computers have adopted metaphors such as the pen and notepad metaphors chosen for GO's PenPoint operating system (Carr 1991) or Apple's Newton MessagePad.

2.4 Mixing metaphors

2.4.1 Mixed metaphors in language

Metaphor can exist at many levels in language. I have already used the example of the central family in 'The Manchurian Candidate' as being analogous to the Kennedys but the same book also includes descriptive metaphors entirely unrelated to this. Consider the sentence, "He felt like dropping the phone, the call, and the whole soggy, masochistic, suicidal thing in the wastebasket." (Condon 1973, p.10). The overall sentence is a metaphor – dropping the whole affair into the wastebasket – for abandoning a particular course of action. Within this metaphor, the action itself is described by a number of metaphors – 'masochistic', 'suicidal' and 'soggy' – to put over the character's feelings about the position he is in.

Most of these metaphors are not related to one another, nor are they related to the higher level political metaphor embedded in the full narrative. Although it may seem like a conflict to describe something as masochistic and suicidal on the one hand and soggy on the other, most of us will have no problem with a sentence like this, when reading it in context, as the metaphors relate to different levels of the narrative. 'Masochistic' and 'suicidal' refer to the possible outcomes of the action, whereas 'soggy' describes the character's ambivalence about acting.

Similarly, unrelated metaphors associated with different aspects of computing can be successfully presented to the user simultaneously, provided they apply to different levels of the presentation. This is quite different from combining aspects of the same overall metaphor, such as 'documents' and 'folders' within a single level.

2.4.2 Mixed metaphors in computing

Foley et al. talk of the different levels at which a user interface can be viewed in linguistic terms - semantic, syntactic and lexical levels, which they equate to functional design, sequencing design and binding design, respectively (Foley et al. 1990, p.394-95). They place metaphor within a higher level than these three, which they call 'conceptual design'. However, the authors go on to contradict themselves, describing metaphor-based design concepts, such as 'direct manipulation' or 'windows' (Foley et al. 1990, p.397,439) which they place at quite different levels of design.

Hutchins (1989) proposes three categories of metaphor in the process of human-computer interaction:

1 Activity metaphors refer to the user's highest levels goals, such as writing a paper, playing a game, or communicating with another person.
2 Mode of interaction metaphors refer to the relationship between the user and computer.
3 Task domain metaphors provide a structure for understanding a particular task.

Although Hutchins gives no empirical underpinning to these categories, they provided an initial framework for the work here. However, initial examination of his system and attempts to fit known metaphors into it shows some inherent problems. Where activity and task-domain metaphors apply to computing this is in the sense of where the metaphor is taken from, rather than what it is applied to. For example, he includes activity metaphors such as 'playing a game' or 'writing a paper', and task-domain metaphors that provide a 'structure for understanding a particular task'. By contrast, Hutchins divides mode-of-interaction metaphors into four modes, each a style of human-computer interaction: conversation, declaration, model world, and collaborative manipulation. It appears that his activity and task-domain metaphors are classified by what a metaphor addresses in the user domain (the vehicle), whereas mode-of-interaction metaphors are classified by how they are applied in the computer domain (the tenor).

In examining examples of metaphors in literature or in use and attempting to find suitable categories, I have produced two separate classification systems, one based on the metaphor's tenor, the other on its vehicle. Like Hutchins', they are experiential rather than theoretical in nature, pragmatic rather than empirical, but they help to show the potential for the use of metaphor in computing.

2.4.3 Categorising computing metaphors by tenor

These categories are based on the tenor, i.e. the aspect of computing a metaphor supports, rather than the origin of the vehicle. The list is not exhaustive but provides a way to see how metaphors currently in use can be mixed.

Concept: Computer as theatre, interface as facade.

A conceptual metaphor provides a way of looking at the entire design process. For example, Laurel (1986; 1993) has advanced the idea of treating the computer as theatre. Her concept does not imply that the system should look like a theatre but suggests ways of structuring the interaction to 'maintain mimesis', keeping the user's interest and attention. Hooper (1986, p.13-14) prefers an architectural metaphor, with the screen acting as a facade that should invite the user to enter.

Design: Using metaphor as a 'tool for thought'.

Design metaphors are used as a 'tool for thought' (Smyth 1995a) and are not necessarily embodied in any part of the final implementation. This type of metaphor often comes from 'brainstorming', with designers and users generating as many metaphors as possible to help provide insights into the design process. For example, the 'Quick-cash' option to obtain £50 from a single action at an ATM (Automatic Teller Machine) came from a brainstorming session which raised consideration of the 'Less than 10 items' tills at supermarkets (personal report from a member of the design team at RACE Concertation Meeting 1993).

Development: Work-flow, system life-cycle, object-oriented design.

These are metaphors employed as part of the design methodology to assist developers in the development process which, again, the user will not be directly aware of.

Hardware: Notebook, notepad, pen, organiser.

The physical packaging of the computer can embody metaphors which might influence the presentation metaphor (see below) but do not necessarily do so. For example, many 'notebook' computers still use 'desktop' interfaces.

System: Directories, menus.

System metaphors describe the internal software structures introduced to assist the user and to structure basic interaction with the system. They are not dependent on the hardware and are often independent of the metaphors for presentation and interaction.

Presentation: Documents, filing cabinets, rooms.

Based on the lower level of Hutchins' classification system, three styles of presentation metaphor can be identified – interactional, spatial and activity-based. These will be looked at more closely later in this thesis.

Interaction: Direct manipulation, command, conversation.

This is independent of the presentation metaphor. For example, the concept of 'moving' a file from one directory or folder to another can be the typed command, 'move', or the dragging of an icon in direct manipulation. It is also independent of the system metaphors – a menu can be directly manipulated as with pop-up or pull-down menus or it can be a menu of commands as part of a command-based system.

Support: Interface agents, speech bubbles.

Metaphors can be used to help the user to understand or use the system. Examples include the 'speech balloons' in the standard Macintosh Help facility and interface agents such as the talking paper clip in Microsoft Office or the 'Coach' in WordPerfect. Additional support metaphors may also be introduced in supporting documentation or third-party manuals, although these are not part of the computer system and will therefore not be considered further.

Different metaphors can co-exist, separately, at each of these levels. Although some metaphors could be difficult to reconcile, metaphors in each category often come from very different sources. Take, for example, an Apple PowerBook running Macintosh System 8:

Table 2.1: Metaphors in the Apple PowerBook.

Metaphor Category	Implementation
Concept	Computer as appliance (the original Mac concept)
Design	not known
Development	Objects, classes, inheritance
Hardware	Notebook
System	Windows, menus
Presentation	Desktop
Interaction	Direct manipulation
Support	Balloon help

Other writers have used the term 'interface metaphor' to refer to any metaphor involved in the user's interaction with the computer software. Although most closely corresponding to the 'presentation metaphor', this could apply to most of the above categories apart from design and development. As much of the literature uses this terminology, I will also use it where distinctions between metaphor categories are not immediately important. I will now examine types of interface metaphors and the categories of metaphor vehicle which they employ.

2.4.4 Categorising interface metaphors by vehicle

The above categories apply to the aspect of computing the metaphor supports (the tenor), but metaphors can also be categorised according to their origin (the vehicle). As with the previous categorisation, Hutchins' (1989) categories do not correspond to the metaphors found in current use. Examination of the metaphors discussed in the two sections above shows that they fall into one of five categories:

1 Spatial metaphors (room, landscape, space)
2 'Communications link' metaphors (net, web, highway)
3 Book or document metaphors
4 Guides or agents
5 Tools (video recorder, pen)

Looking a little deeper, it is apparent that the spatial metaphors depend heavily on structuring information. Both communications and book metaphors concentrate on the linking and interaction between units of information. Despite their immediate differences, it therefore makes sense to group these two together as 'interactional metaphors'. Agents and tools can also be grouped: they do not structure the information or provide links between information units, but instead provide an intermediary to allow the user to carry out specific activities. The five categories can therefore be reduced to three basic forms:

1 Spatial metaphors
2 Interactional metaphors
3 Activity-based metaphors

Aspects of each property can be reflected in a single interface metaphor but stressing one aspect of the metaphor will tend to decrease the extent to which the other properties apply. We can therefore consider any metaphor-based interface as a point somewhere within a triangular area:

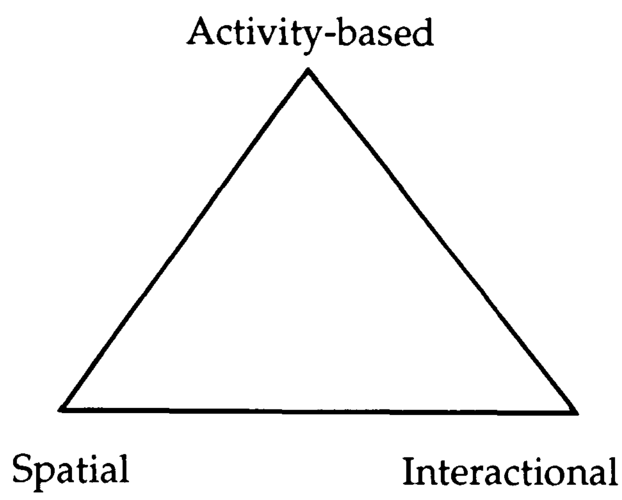


Figure 2.4: Three qualities of an interface metaphor.

Of the three, spatial metaphors have probably been given the greatest attention. Some have argued that they are inherently better than other types of metaphor:

> The special cognitive reality of space ... makes the spatial domain particularly suitable as a medium for conveying knowledge, since its properties are universal to different cognitive systems. Thus, the spatial domain can be used particularly well as the source domain for metaphors with a non-perceivable or abstract target domain. In this way, the properties of physical space can be used as vehicle for conveying non-spatial concepts...
>
> I propose that our knowledge about the organization of space serves as a "cognitive interface" between abstract and non-perceptual knowledge and the "real world". In other words, we may interpret non-spatial concepts by mentally transforming them into spatial concepts (i.e. understanding them in terms of spatial concepts), carrying out mental operations in this "visualizable" and "graspable" domain and transforming the results into the original domain.
> (Freksa 1991, p.362).

It should be noted that this classification deals with the metaphor vehicle, not the medium in which the interface is presented. For example, adventure games and MUDs (Multi User Dungeons) often use text to create spatial environments, as in, "You are in a room. There are exits to the North, South and West. In front of you is a staircase." Text-based spatial metaphors have also been used to develop more 'serious' interfaces, such as Dieberger's Information City (Dieberger 1994a; 1994b). Some have even developed auditory environments based on spatial metaphors (Lumbreras 1993; Mynatt 1994).

Rather than the medium, therefore, the three categories of metaphor depend on how they support the user. Spatial metaphors structure people and information according to *where* they are, activity-based metaphors by *what* can be done in relation to the information or the people, and interactional metaphors by *how* the units of information, or people, link to one another.

2.5 Similar interface concepts

There are some concepts used in the design of computer interfaces which are superficially similar to metaphor but depend on different processes. Although separate from metaphor, many of them will feed into the development of metaphors.

2.5.1 Puns

Whereas metaphor is a matter of taking a concept from one domain into another, a pun depends purely on word play. Puns depend on the specific language they are expressed in and can be distinguished from metaphors by changing the language. For example, the program 'MicroPhone' is used to connect a micro (computer) to a (tele) phone and uses a microphone as its icon. Although the French language uses the same term *microphone*, this does not obviously relate to connecting an *ordinateur* and a *téléphone*. A true metaphor, such as 'move', will still work when expressed as *remuer*.

2.5.2 Anthropomorphism and animism

Anthropomorphism is used to describe the 'humanising' of non-humans, whether animals or objects and is reflected in product names such as 'Mr' Sheen, the Sony 'Walkman' or 'Gameboy'. An example of anthropomorphism in computing is the 'smiling Mac' that appears at start-up on the Macintosh, or the 'sad Mac' when the computer cannot start.

The term 'animism' is used in two contexts. In developmental psychology the phrase is used to describe a phenomenon in which children credit inanimate objects with self-will. Animism is also used to describe the presence of a human-like spirit inside an object, such as water nymphs or wood nymphs. In anthropomorphism, external human attributes are imposed on an object, such as people's names or faces; in animism the human attributes are internal, 'waiting to get out'. The following quotation makes this distinction clear in the context of computing:

> Computer scientists have tended to shy away from personifying machines, but we felt we were seeing a call for it from users. We were reminded of the reactions to Weizenbaum's psychologist program ELIZA in the 1960s. Some users actually sent observers out of the room because they were having a private conversation, though they knew their partner was a computer. Also, we believe there is a difference between portraying characters within the database versus anthropomorphizing the machine itself. The projection that occurred within ELIZA was not "a computer is a person," but rather "there is a doctor in the machine." Similarly, none of our users said *the computer* is betraying me or *the computer* is mad at me. Rather, the relationship occurred between the user and the image of the guide.
> (Oren 1990, p.373)

Although Oren et al felt they saw a call from users for personification, other researchers have found serious problems with it. Quintanar et al (1982) examined responses to an anthropomorphic dialogue, comparing it with a mechanistic dialogue and found that users considered the anthropomorphic design to be 'less honest'. Other research has been ambivalent, finding a degree of chattiness to be beneficial but that users are quickly bored if it becomes excessive (Spiliotopoulos 1981).

2.5.3 Personification (or prosopopeia)

This is a trope, rather than a psychological phenomenon, and includes both the animistic and anthropomorphic principles. It can be applied to abstract concepts as well as concrete objects and is commonly used in romantic or detective fiction with expressions such as 'Fortune smiled on her that day' or 'he looked into the eyes of Death'. The scope for personification in computing is small, except in the sub-forms of animism and anthropomorphism described above.

2.5.4 User-friendliness

The concept of user-friendliness has been condemned by Shneiderman (1987, p.73) as a 'vague and misleading notion'. It is a special case of the anthropomorphic and animistic principles, applying the human quality of 'friendliness' to a machine or program. Take, for example, the following message from the Macintosh version of Eudora (an Internet mail program):

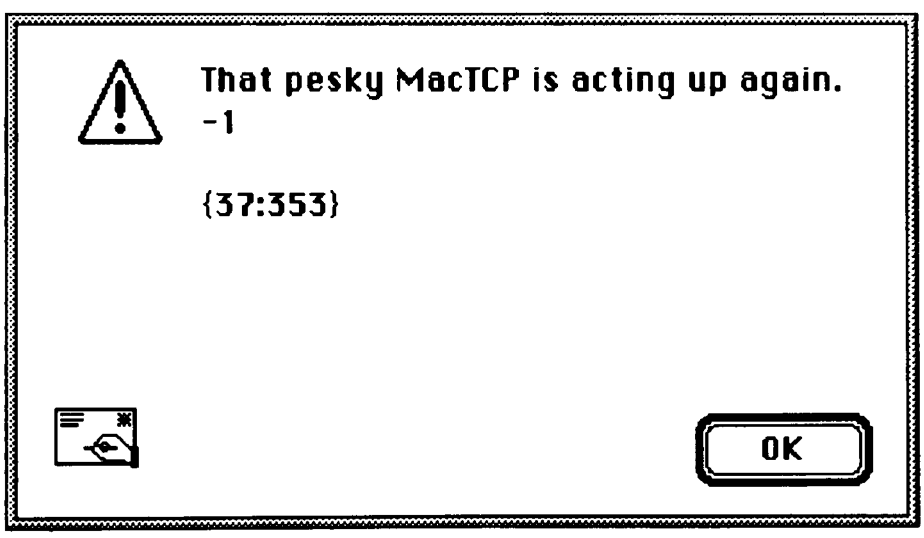


Figure 2.5: An example of 'user-friendliness'.

The Eudora message is chatty in form, implying an ability to converse in a human manner. As Schneiderman (1987, p.323) points out, "Attributions of intelligence, independent activity, free will, or knowledge to computers can deceive, confuse, and mislead users." This is demonstrated by the error codes '-1' and '{37:353}' in the message above which are completely impenetrable to most users. Had the program been truly 'knowledgeable' and 'intelligent', it would have realised that, in this case, the modem was switched off.

2.5.5 Literal categories

Some people will perceive a concept as a metaphor that another will simply regard as a valid sub-category of the word. Goatly (1997, p.21) points out that phrases such as 'a pike is a kind of fish', a literal categorisation, are very similar to metaphors such as 'a sock is a kind of glove' and that some phrases, such as 'an escalator is a kind of staircase', are ambiguous. An example in computing might be 'directory' which Chambers English Dictionary (1990) describes both as a book and as 'a body of directions'. Although a disc directory is literally 'a body of directions', all pre-computing directories existed on paper and many users would see this as a metaphor.

2.6 Benefits and dangers of computing metaphors

2.6.1 Metaphors and the learning process

Most writers in the past have emphasised the value of metaphor to computing, particularly as part of the learning process. The apparent advantages of interface metaphors are relatively simple to explain:

> An alternate approach to controlling the complexity of user interfaces is to design interface actions, procedures and concepts to exploit specific prior knowledge that users have of other domains, for example, to design an office information system using the metaphor of a desktop. Instead of reducing the absolute complexity of an interface, this approach seeks to increase the initial familiarity of actions, procedures and concepts that are already known. The use of interface metaphors has dramatically impacted actual user interface design practice. (Carroll 1988, p.67).

In an earlier paper, Carroll, Kellog and Mack (1985) had put this in terms of 'active learning', claiming: "Metaphors can facilitate active learning...

by providing clues for abductive and adductive inferences through which learners construct procedural knowledge of the computer." Cornell Way agrees with this view: "Metaphor is important because it provides us with a way of moving from known ideas and familiar concepts to new and unknown ones ... Metaphor then is important to learning; it is easier to take parts from other established concepts than to build up new ones from scratch." (Cornell Way 1991, p.8).

In parallel work, looking at the use of the typewriter analogy for text editors, on this occasion by people learning about them, Douglas and Moran came to similar conclusions. They invoke the concept of problem space to explain the process:

> The user is trying to acquire the cognitive skill required for expert use of a text editor. Text editing skill can be represented as a *problem space*. The initial task is to build such a problem space. This is done incrementally, not by some sort of pure induction, but rather by borrowing skills from other related domains, which we also consider to be represented by problem spaces. (Douglas 1983, p.207).

2.6.2 Metaphor fit

Some authors have identified problems that metaphors in computing can cause, particularly where the fit, or tension, between the vehicle and the tenor is poor. Where the metaphor is not explicit and the basis of classification is unclear, the scope for semantic confusion in the user becomes greater. Carroll and Thomas (1982) discuss a number of examples, while another study by Carroll and Mack (1984) looks at the particular problems of people learning to use standard word processors. In this study they discovered that many problems were caused by the users' expectation that the word processor would behave like a typewriter. For example, users expected the text to move up the page when they pressed the return key as would happen when pressing the carriage return of a typewriter. In a later paper, Carroll and Mack point to the inevitability of mismatches:

> Metaphors, by definition, must provide imperfect mappings to their target domains. If a text-editor truly appeared and functioned as a typewriter in every detail, it would be a typewriter. The inevitable mismatches of the metaphor and its target are a source of new complexities for users. (Carroll 1988, p.69).

Although mismatches may be inevitable, not all authors see them as necessarily wrong:

> These mismatches of metaphors often are important factors of the force of the metaphor. Mismatches in the metaphor can help considerably making a system useful if the mismatches are designed well. The user interface principle of forgiveness is particularly important in metaphor mismatches - it allows the user to explore those unfamiliar features of the system and by exploring them she easily learns to use them for her own benefit. (Dieberger 1994a, p.57)

Hammond and Allinson (1987) have criticised interfaces that are too heavily dependent on metaphors and Johnson gives some examples of the problems this could cause:

> An exact simulation of a book on a computer would force the user to slowly turn one page after the other... imagine an online documentation system that displays a document as a book; users display the next page of text by pointing (with the mouse) to the corner of the page, depressing the mouse button to 'grab' the corner, and pulling it across the screen to the other side, with an accurate animation of the whole sequence. (Johnson 1987, p.21).

Kay also attacks the too literal implementation of metaphors, calling for greater use of 'magic':

> For example, the screen as "paper to be marked on" is a metaphor that suggests pencils, brushes and typewriting. Fine as far as it goes. But it is the magic – understandable magic – that really counts. Should we transfer the paper metaphor so perfectly that the screen is as hard as paper to erase and change? Clearly not. If it is to be like magical paper,

> then it is the magical part that is all important and that must be most strongly attended to in the user interface design. (Kay 1990, p.199).

Sometimes 'magic' is explicit, as in games that use 'teleport' devices. Other examples are more mundane, such as the ability to paste an unlimited number of times with the 'cut-and-paste' metaphor. For example, Dieberger explores the many magical features of the Macintosh 'folder' metaphor (Dieberger 1994a, p.60).

Whereas magic adds features, designers also leave out features not relevant to computing activities. For example, gravity is useful for the organisation of a physical desktop but also means that things can fall off onto the floor; on the Macintosh desktop things simply stay wherever they are placed. "In an actual interface design process, the designer has to decide which features of a source domain are to be considered salient and which are not." (Kuhn 1991, p.423). Having made these decisions, it is also critically important that the designer communicates this information to the user:

> "Magic features" have implications on how the system should be taught to users. For example it should be made clear to users of a computer file system that the system is not "just like" a file cabinet in an office but provides additional functionality. Otherwise users may believe that limitations and other irrelevant aspects of the physical file cabinet apply also to the computer system and is confused by the "magic features"....
> Magic features can make metaphors much more useful but they must be pointed out to users and they should be designed in a way to make their working easily comprehensible. (Dieberger 1994a, p.60).

Pointing out the magic features to the user obviously depends on the designer being aware that they are magic. In the example of Carroll et al's typewriter metaphor, discussed above, there had been no conscious intention to use a metaphor on the part of the designers. The problem was that the metaphor was implicit and thus unstructured. Had the designers explicitly used the typewriter metaphor, the system and the training that the users received could have been designed in such a way as to make this clear and, most importantly, to make clear where the interface diverged from that of a typewriter.

2.6.3 Implicit and explicit metaphors

This example suggests that it is impossible for interface designers to avoid metaphor. If designers attempt to build metaphor-free systems, either they will introduce metaphors subconsciously or the users themselves will introduce them. As designers can have no control over metaphors they are unaware of, the effects of the metaphor on the users will be unpredictable, subsequent revisions of the software may not take the metaphors into account and, at worst, they will lead to confusion on the part of the users. Under these circumstances, it is probably better for the designer to explicitly choose appropriate metaphors and keep control of them in the development of the software.

Carroll and Mack's 'typewriter' example was implicit for both the designer and the user. Problems can also occur when the designer's explicit metaphor is not recognised by the user. Anyone who has helped naive users has probably come across examples of this type, particularly where an icon has been mistaken for a picture of something it is not intended to be.

Problems of this type may be seen as an example of cultural mis-match between worlds of the designer and the user, such that the designer is not conscious of the metaphor or, if conscious, does not realise its potential impact on the user. Some extreme examples of cultural mis-matches are given by Grundy (1996, p.85-94). Her main argument is that, "Computing is taught using metaphors, analogies and examples drawn largely from a male environment. Women students have therefore always been required to understand what they are taught through a screen of male values and experience." (Grundy 1996, p.88).

She reserves her heaviest attack for 'rape metaphors' in computing, quoting Francis Bacon's description of scientific investigation, in which he refers to nature as 'her': "Neither ought a man to make scruple entering and penetrating into those holes and corners when the inquisition of truth is his whole object". She argues that "Bacon's rape metaphors helped to shape the methodology of science powerfully in the past... Although the notion of rape itself may have disappeared, the language and imagery of male-dominated sex and violence still persists in computing jargon and must influence girls and women, particularly as they start to learn the subject". She then gives examples of such terms including 'violation', 'degradation', 'chaining', 'abort', 'kill' and 'execute' (Grundy 1996, p.90-92).

The importance of Grundy's argument does not depend on any underlying validity. The fact that she is sincere in her views and that other women may share her viewpoint means that they will feel alienated by metaphors which they see as degrading towards women. The casual use of the term 'abort' in computing is an example that many people might appreciate. Although 'abort' has the wider meaning of abandoning an action, in popular use it almost always refers to the abortion of a pregnancy. A comparable 'male' term might be to replace the error message 'the command could not be completed due to insufficient memory' with 'your command was impotent due to insufficient memory'. This would be technically correct but many men might feel uncomfortable with it.

2.6.4 Opposition to interface metaphors

One of the strongest attacks on interface metaphors comes from Nelson (1990, p.236) who identifies three 'elements of bad design', one of which is what he terms 'metaphorics'. "I would like to venture that this 'metaphor' business has gone too far. Slogans and catchphrases are all very well, and these things have their uses for people who are going to learn software *approximating* rather than *understanding*.". He also claims that, "the metaphor becomes a dead weight," and suggests that the "alternative to metaphorics is the construction of well-thought-out unifying ideas, embodied in richer graphic expressions that are not chained to silly comparisons." (Nelson 1990, p.237). His alternative is 'the design of principles', giving VisiCalc as an example and suggesting the use of hypertext as a future alternative (Nelson 1990, p.240-242).

Although metaphor might be a 'dead weight', his examples of better practice do not appear to provide a metaphor-free alternative. VisiCalc may have been a new concept for most of its users but is basically a metaphor, based on spreadsheets of a type already in manual use by accountants. Hypertext is undoubtedly a useful tool, but the references given in Section 2.3.2 show that many developers find it is only part of the answer and are searching for suitable metaphors to help prevent users becoming 'lost in hyperspace'.

Alan Kay (1990) also offers a critique of metaphor, "My main complaint is that *metaphor* is a poor metaphor for what needs to be done. At PARC we coined the phrase *user illusion* for what needs to be done." He continues, "it is the magic... that really counts." (Kay 1990, p.199). The importance of magic has been acknowledged above, but it depends on an underlying metaphor to be seen as magic. He offers examples of 'better' solutions taken from interfaces such as Smalltalk and TEX (Kay 1990, p.200; p.203). Although both systems have their adherents, neither has become as universally accepted as the metaphor-based interfaces Kay rejects. Popular acceptance should not, of course, be taken as an automatic guarantee of usability, but requires an explanation if his argument is to be accepted. As a future direction, Kay (1990, p.205-07) offers agents as a more promising possibility, although it can be argued that these are only a particular type of metaphor.

Both of these critiques come from a series of what the editor, Laurel (1990 p.187), introduces as 'sermons' rather than academic studies and offer the authors' personal views rather than experimental results. In general, academic work, such as that by Carroll cited above, has shown that metaphors are unavoidable, being introduced subconsciously by the designer or consciously by the user. However, criticism of the use of metaphor or on the specific choice of interface metaphor, such as those by Grundy and Nelson, suggest that metaphors can raise significant emotions and signify far more than a simple aid to the learning process. I will examine this signification further in the next chapter, looking at whether semiotics, the study of signs, might assist the interface designer to a more complete understanding of human-computer interaction, in particular the role of interface metaphors.

3 A semiotic model of HCI

3.1 Introduction

Chapter 2 referred to the field of semiotics. It was first mentioned in Weinrich's definition of a language "different from other semiotic mechanisms" (Weinrich 1966, p.142). Although Weinrich's very strict definition of a language could not be taken to include GUIs, it could be that they form another type of semiotic mechanism. Also, when considering the evolution of the desktop metaphor, reference was made to Nadin's work on the Lisa, in which semiotics was used to structure the interface design (Nadin 1988). In this chapter I will follow this up with a consideration of whether semiotics provides an effective tool for the analysis of interface metaphors.

Semiotics began at the end of the last century, when de Saussure proposed the creation of a new study he called semiology (now more usually known as semiotics), "a science that studies the life of signs within society", of which "linguistics is only a part." (de Saussure 1974, p.16). Research into computer interfaces, including GUIs, inevitably centres on the study of 'signs within society' and it would thus seem that semiotics offers a potential discipline for achieving a better understanding of the way in which the user and the computer interact, including the role of metaphor in their interaction.

3.2 Semiotics

3.2.1 Background

Although semiotics is comparatively recent, it rests on thousands of years of the study of rhetoric. Classical rhetoricians such as Aristotle and Quintilian categorised the methods and tools of description and argument within highly stylised forms of language such as poetry and drama. They identified a rhetorical discourse as consisting of "invention" (developing arguments), "disposition" (organising the subject), and "style" (the means of persuasion). Two types of stylistic device, or rhetorical figure, were identified:

1. Schemes, which deviate from normal language mainly in the selection of vocabulary or ordering of words.
2. Tropes, which deviate from common usage mainly in the meaning of words.

Much attention was given in earlier research on computer languages to their schematic structure or syntax, comparing, for example, the postfix syntax of languages such as Forth with the use of the prefix syntax common in functional languages. In contrast, this thesis concentrates on tropes, principally metaphor. As discussed in Chapter 2, the choice of syntax can be influenced by the metaphor of interaction but much of the importance of metaphor lies in its status as a trope, which is independent of the syntax or even the mode of presentation.

Having been developed by ancient Greek and Latin philosophers, the application of rhetorical analysis was mainly restricted to those languages, particularly to formalised language such as poetry. Large numbers of schemes and tropes were identified by early rhetoricians but most subsequent work consisted of categorising the descriptive structures of poetry and other texts, rather than developing a deeper understanding of why or how they work. Rhetoricians and grammarians lost touch with contemporary language except in etymology, the derivation of words.

Even this was heavily influenced by the desire to ensure 'correct' usage of spelling and syntax by reference to historical roots.

This approach was challenged by de Saussure who began to examine the structure of contemporary *parole* (language as actually spoken by people), rather than the formal *langue*. He defined his approach as *synchronic*, looking at the full structure of language at a given time, in contrast to the dominant *diachronic* approach of etymology, looking at small elements of language as they change over time (de Saussure 1974, p.81). In doing so, de Saussure is widely recognised as the first proponent of what became known as structuralism, the synchronic approach that dominated linguistics, and influenced many other studies, through much of this century. Piaget (1971), one of structuralism's leading exponents presents a good introduction to leading figures in the movement while Sturrock (1986) provides a more recent overview. While structuralism has been successfully applied to many social sciences and to mathematics (Piaget 1971, 17-36), Chomsky has proved the most important structuralist in developing de Saussure's work in linguistics – see Chomsky (1975; 1986) for overviews.

De Saussure saw his new approach to linguistics as part of a greater science, the new science of 'semiology' – the study of signs:

> *A science that studies the life of signs within society* is conceivable; it would be part of social psychology and consequently of general psychology; I shall call it *semiology*..... Linguistics is only a part of the general science of semiology; the laws discovered by semiology will be applicable to linguistics, and the latter will circumscribe a well-defined area within the mass of anthropological facts.
> (de Saussure 1974, p.16).

At the same time, the philosopher Peirce (1985) was also setting out the boundaries of a field of study that he termed 'semiotic'. Both adopted the prefix 'semio' from the Greek for sign, σεμειον (semeion). Whereas de Saussure saw 'semiology' as a branch of psychology, Peirce termed the field 'semiotic' which he saw as a branch of logic. It is now usually known as semiotics, analogous to its major component linguistics. Although each approached their studies from different viewpoints, one central issue united de Saussure and Peirce: a sign cannot be studied without considering what it signifies to a person.

3.2.2 Terminology

The somewhat tautological definition of a sign in semiotics is 'anything that signifies something to someone'. The concept of a 'sign' can thus include a word, a sentence or an entire book. It also includes non-verbal signs: a statue, a diagram or a photograph. A sign does not have to be man-made: clouds on the horizon might signify rain; spots might signify measles. Various terminologies have been used to express the concepts involved. I have adopted de Saussure's original terminology (de Saussure 1974), with additions by Eco (1979). In this, the perception of spots forms the **signifier** and the concept of measles is the **signified**. The third element is the act of **signification**, which is context dependent: a modern doctor might see spots as signifying a disease, but other societies might interpret them in very different ways.

All three of these terms describe mental constructs: the spots might be seen quite differently by someone who is colour-blind, while a visual disorder might lead to someone seeing spots with no physical existence. Reserving the term **sign** to refer to the physical sign, I will use Eco's term **sign-function** to refer to the total conceptual system as shown in the following diagram based on that of Eco (1979, p.58):

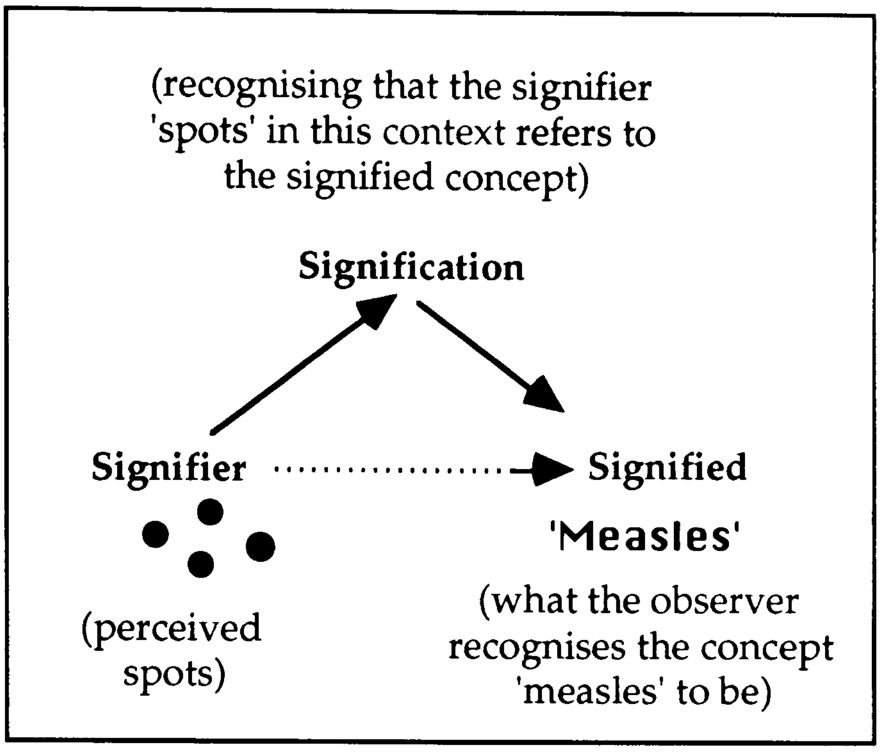


Figure 3.1: The sign-function.

The original form of this diagram comes from Ogden and Richards (1938) although they used different terms. This and similar versions are discussed by Eco (1979, p. 58-59) and by Sebeok (1991, p.52), who strongly disagree on its value. Whereas Sebeok sees the triangle as "heuristically valuable", Eco attacks it as, "an over-simplified diagram which has rigidified the problem in an unfortunate way." It should be taken here as a simplified heuristic aid, rather than an all-encompassing, rigid definition of a sign. Martin (1975, p.26) uses similar terms but extends the triangle to a rectangle to include the physical sign.

As the triangle shows, the sign only carries meaning as part of the process of signification. It is an interactive act, depending on the way it is perceived rather than the intent in generating it. Semioticians usually express this as 'reading' meaning into the sign, referring to any related collection of signs as a 'text'. Despite the terminology, these terms are not limited to writing and semiotics is applied to 'reading' non-verbal 'texts' such as a film or the contents of a painting.

Semiotics also takes great note of what is termed 'intertextuality' (Chandler 1995). We can never read a text in isolation; it must be read in the context of other texts of the same type. In the case of computing, we understand a particular Macintosh wordprocessor in the context of other wordprocessors and other Macintosh programs we have used. Applying this to interface design implies that interfaces should be consistent across applications and across platforms. As this principle is already commonly held in HCI, it will not be pursued further. Instead I will look a little more closely at metaphor in semiotics.

3.3 Two types of metaphor

3.3.1 Poetic metaphor

Metaphor can be seen in two different ways. Either a new idea is created from the fusion of the two original ideas, or our understanding of the first idea, or tenor, is transformed by consideration of the vehicle. These can be represented symbolically as:

(1) $T + V \Rightarrow C$, or

(2) $T + V \Rightarrow T(V)$,

where T is the tenor, V the vehicle and C is a new concept created by the use of the metaphor.

The role of metaphor, and that of metonymy, in the development of language and computer interfaces was discussed in Chapter 2. I hope to establish that this is through process (1) above and that this is also the process of most importance in computing. Much work on metaphor can be disregarded as concentrating on poetic metaphor which is represented by process (2), where the vehicle is introduced to change one's perception of an existing concept - the tenor - rather than create a new concept.

The most extreme case for the role of metaphor in our language is that presented by Lakoff and Johnson (1980), as introduced in the previous chapter. After stating that, "Our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature," they continue:

> The concepts that govern our thought are not just matters of the intellect. They also govern our everyday functioning, down to the most mundane details. Our concepts structure what we perceive, how we get

> around in the world, and how we relate to other people. Our conceptual system thus plays a central role in defining our everyday realities. If we are right in suggesting that our conceptual system is largely metaphorical, then the way we think, what we experience, and what we do every day is very much a matter of metaphor.
> (Lakoff 1980, p.3).

There is some experimental evidence to support at least some of this viewpoint. If most of our language were literal, with metaphor as an 'add-on', this would imply that we would comprehend the literal meaning of an expression and then work out the metaphor, taking longer than the interpretation of a purely literal text. In fact, evidence generally supports the view that metaphors take no longer to comprehend than literal expressions. See Cornell Way (1991, p. 51-59) for summaries of a number of experiments which have shown equivalent comprehension time for metaphorical and literal expressions.

Some evidence has been presented to oppose this view which has shown that it is possible to 'force' a longer reaction time by leading the reader "down the literal path" (Gerrig 1983, p. 668). For example, it takes longer to interpret, "The concert hall was filled with sunshine by the orchestra," than, "The orchestra filled the concert hall with sunshine". Gerrig and Healy (1983) contend that we prefer to follow a literal reading first but that a literal interpretation of the vehicle is quickly truncated by the introduction of the tenor. However, their examples do more than their claim to "lead subjects down the literal path", they actually introduce greater ambiguity into a half-read sentence which the reader must interpret. Consider the number of potential literal mis-interpretations in the two examples they use:

[[[The concert hall was filled] *(i.e. a full house)* with sunshine] *(i.e. it was well-lit)* by the orchestra] *(i.e. the orchestra generated happiness).*

[[The orchestra filled the concert hall] *(i.e. occupied the concert hall)* with sunshine] *(i.e. the orchestra generated happiness)..*

The first form of the sentence does more than guide the user to a literal interpretation; it introduces greater ambiguity that will take longer to resolve. It may also be argued that comprehension of the active form is faster than that of the less common passive form. It is therefore not surprising that reaction times are greater in this case. Other experiments have looked at the effect of context, finding that more contextual information, which reduces ambiguity, leads to no difference in comprehension time between metaphor and literal expressions (Ortony 1978). A summary of research in this area by Hoffman also appears to confirm Lakoff and Johnson's views:

> In ordinary contexts, figurative language takes no longer to comprehend than ordinary communication, because figurative language *is* ordinary communication. It does not seem to require special comprehension processes, if to be "special" means "to take more time". (Hoffman 1984, p.154)

However, Levin (1988) has argued that Lakoff and Johnson's work is questionable in treating almost all language as metaphorical in that many of their examples have become entirely lexicalised and that we are no longer aware of their metaphorical nature. He argues that they are conventionalised metaphors that do not demand that we think of concepts in a new way. But, he claims, this is exactly what poetic metaphor does, causing us to 'imagine a metaphoric world in which trees actually do weep'.

Levin's claim that language is not full of poetic metaphor can be accepted but does not mean that language is not full of metaphor. The metaphors that Lakoff and Johnson or Eco deal with are not used for poetic effect but as part of the basic structure of language and thought. Poetic metaphors are used to modify existing concepts, whereas scientific metaphors and the metaphors employed in language development are used to create new concepts. To distinguish this type of metaphor from poetic metaphor, I will refer to it as *generative metaphor*.

3.3.2 Distinguishing poetic and generative metaphor

When using computers, we are dealing with concepts not previously held in our language. Metaphor, here, applies a **familiar** vehicle to an **unfamiliar** tenor. Applying the vehicle within the context of the computer system, we generate an entirely new concept, as in process (1), the generative metaphor:

(1) T + V => C

Poetic metaphor differs from this in offering the transformation of one concept by linking it with another: a **familiar** tenor and a **familiar** vehicle. In picturing the two concepts together, the tenor is enriched by its association with the vehicle, the poetic metaphor:

(2) T + V => T(V)

Or, as Martin puts it:

> When Lowell writes of 'yellow dinosaur steamshovels', the actual appearance of dinosaurs and steamshovels has to be contemplated and compared in imagination. We have to picture both tenor and vehicle, and fit them over each other, 'picturing' both at once.
> (Martin 1975, p.209).

Another expression of this idea is provided by Hester (1967). He stresses the superimposition of ideas in the experiencing of metaphor, but uses Wittgenstein's discussion of *seeing as* in *Philosophical Investigations* to explain his views. Hester writes: "Metaphor involves . . . the intuitive relation of *seeing as* between parts of the description . . . [it] involves not only a tenor and vehicle, to use Richards' terms, thrown together in a sentence, but the positive relation of *seeing as* between tenor and vehicle."

Empirical evidence for this viewpoint is given by the work of Kelly and Keil (1987), who examined whether, "comprehension of a metaphor alters one's understanding of a domain over and above the concepts explicitly stated in the metaphor." They conducted a series of experiments in which they found that comprehension of the metaphors not only increased the similarities between the tenor and the vehicle but also increased similarity between other concepts from the same domain which could have formed different but appropriate metaphors if related.

By contrast, terms from the two domains that would form inappropriate metaphors if related tended to decrease in similarity. They concluded:

> ...first, that whole domains of concepts are implicated immediately in the process of comprehending individual metaphors. In addition, the conceptual domains interacting in metaphor are restructured, at least in terms of the similarity relations between concepts within the domains. Finally, this restructuring is asymmetric in that the tenor's domain undergoes greater change than the vehicle's domain.
> (Kelly 1987, p.47).

This supports the case that, in the case of poetic metaphor, the complete mental model of the vehicle is applied to the tenor: the two are mentally 'superimposed', modifying the tenor rather than generating an independent concept.

3.3.3 Poetic metaphor in computing

Compared to the history of language, the history of computing is extremely short. Generative metaphor thus plays a major role in providing expressions for the new concepts which computing has introduced. By contrast, poetic metaphor depends on the application of a familiar vehicle to a familiar tenor. Few concepts in computing have yet become familiar enough for poetic metaphor to be used but an obvious exception is the computer itself. An example of poetic metaphor has been the naming of certain types of computer (and sometimes computer terminals) as 'workstations'. Chambers dictionary lists the principal meanings of 'work station' as "a position at which particular work is done." (Schwarz 1988). At the time that the term came into general use, computers were classified as 'mainframe' or 'mini' computers; rather than the obvious term 'micro computer' for still smaller machines, some manufacturers preferred to drop the term 'computer' altogether and use the new metaphor 'workstation' which drew attention to the new role they foresaw for this type of computer.

A second circumstance in which poetic metaphor can be used in computing is where an additional, poetic metaphor is applied to a concept which has already been identified by a generative metaphor. Thus, Windows 95 includes a special type of folder known as the 'My Briefcase'. The term 'folder' is an example of a generative metaphor with which it is assumed the user has become familiar but the user is then asked to consider one folder as if it is also a briefcase. My Briefcase behaves like a normal folder in most senses but includes an additional 'Briefcase' menu for performing specialist tasks such as updating data on a portable computer. In other words, the naming of the folder as a briefcase has transformed the perception of the folder, modifying the tenor rather than generating an original concept – the definition of a poetic metaphor.

3.3.4 Generative metaphor

At the simplest level, generative metaphors may appear to be what Goatly refers to as "lexical filling" (Goatly 1997, p. 27). This is a simple means by which children learn language. Goatly gives the example of his child talking of the "shell of the bread." He points out that this is not truly metaphor – the word 'crust' has not yet entered her vocabulary and so she uses the word with the 'best fit'. Similar lexical filling happens in adult language when the boundaries of a category are not understood. The 'hedge sparrow' looks like a sparrow but is actually a dunnock; similarly the 'sea anemone' looks similar to an anemone but is actually a sea animal. However, the processes involved in the two cases are quite different: 'sea anemone' is an example of generative metaphor, using a familiar word to name a new object which is clearly not an anemone, whereas 'hedge sparrow' represents an error in categorisation or lack of knowledge of the true name. To refer to a camel as a 'ship of the desert' is different again, involving poetic metaphor: both 'camel' and 'ship' must be understood before the metaphor works.

Generative metaphor plays its most important role when the object or concept is completely new to the culture. Most of the concepts used in computing fall into this category, having been developed within the last 30 years. Also, as explained in Chapter 2, computers have a unique ability to emulate other systems. This power extends to generative metaphors. With natural objects, we might use a metaphor to name them according to their natural appearance, as with the 'sea anemone'. In contrast, computer software and interfaces can be re-structured to fit almost any metaphor we choose. The leaves and branches of a tree will always have a tree-structure but computer 'objects' can be structured as a tree, a chain, a stack or a queue, according to the metaphor we choose to apply. Thus computer interfaces have a greater scope for the use of metaphor than anything previously encountered.

3.3.5 Metaphor and metonymy

As explained above, Lakoff and Johnson placed metaphor as an essential ingredient of cognition. In a later work, Lakoff extends this to cover metonymy, which he sees as "one of the basic characteristics of cognition," in which people "take one well-understood or easy-to-perceive aspect of something and use it to stand either for the thing as a whole or for some other aspect or part of it." (Lakoff 1987, p.77).

As a generative device comparable to generative metaphor, metonymy is certainly very important in naming physical devices, as with the floppy disc or the keyboard, constructing a term from the properties of a device or naming it after its function. In computer interfaces, however, I would assert that metonymy must always be of secondary importance to metaphor, particularly in its ability to help the user understand new concepts. Lakoff writes of "one well-understood or easy-to-perceive aspect of something" but this depends on understanding something or easily perceiving its aspects. If a computing function is entirely new and a new term must be generated then it is unlikely to have any fixed aspects to name it by, unless these are introduced through the mechanism of metaphor.

Consider, for example, the DOS command 'PRINT'. Apart from actually printing by sending output to a printer, the command can be used in many other ways: sending output to a spooler, sending output to a file, sending output to the screen, and so forth. In the strict sense, the name of the command is metonymic: it applies to only one feature (the original function) of the total functionality, but this has happened through the extension of the command from its original function, a process based on metaphor, *seeing* the other functions *as* metaphorically 'printing' to a file, to the screen, and so forth. By naming something after a single aspect or feature, we tend to concentrate attention on that feature. New users of DOS are likely to think that the PRINT command can only be used for printing, limiting the signification of the sign to less than its total functionality.

It is difficult to see any circumstances where metonymy will significantly increase signification as metaphor can. This is because metaphor brings the vehicle – an additional concept – to the tenor whereas metonymy uses an existing aspect of the concept which is already "well-understood." The unique role of metaphor in increasing signification forms a central feature of my semiotic model of human-computer interaction.

Before describing the semiotic model, it is worth considering whether another approach might have provided a more effective route towards building a model of the metaphor process. Semiotics is an example of the linguistic approach considered as a possible approach in Chapter 1. Two other approaches mentioned were computer science and standard HCI techniques. As was explained, the former could provide useful tools in the formal specification of a model, while the latter offers useful techniques for testing the usability of a specific metaphor. However, neither approach provides a suitable theoretic framework for constructing a model of the metaphor process. The fourth approach considered in Chapter 1, that of cognitive psychology, appears to be more promising. It is therefore worth considering what cognitive psychology, or an associated approach, might offer and how this might be compared to the semiotic approach.

3.4 Alternative approaches

3.4.1 Other research into interface metaphors

There is a well established body of research on computer interfaces in the fields of experimental psychology and software ergonomics. However, to date, this work has concentrated on testing the effectiveness of specific interfaces or interface features. The work of Carroll and others, reported in Chapter 2, provides valuable insight into how metaphors affect the usability of specific interfaces but more extensive research on metaphor depends on creating an underlying model of the process of metaphor and testing this model. I propose basing such a model on semiotics, but semiotics is not the only discipline which could be used to build such a model.

Whittock identifies two potential approaches to the study of metaphor:

> Theories of metaphor are closely related to theories of imagination and to the processes and structures imagination employs. The study of metaphor leads off in one direction towards cognitive psychology with its interest in the mental processes underlying perception and mental categorization; in another direction towards rhetoric and strategies of communication. (Whittock 1990)

A third direction, not identified by Whittock, might be to look at metaphor as a part of society. For example, the work of Hutchins, described in this and the previous chapter, is based on his experience as a social anthropologist, but there is not yet any empirical basis for this approach. Research methods in anthropology, such as ethnology, could be used to test Hutchins' ideas but this work is yet to be done. Of the two directions proposed by Whittock, this chapter has thus far concentrated on 'rhetoric and strategies of communication' applied to verbal and non-verbal signs, i.e. semiotics. Before developing my semiotic model, it is necessary to look at ways in which cognitive psychology might provide an alternative route to understanding interface metaphors.

3.4.2 Mental models and HCI

Mental models (Johnson-Laird 1983) have provided one of the most popular tools for cognitive scientists to develop models of human-computer interaction, of which Tauber (1991) introduces a number of examples. A particularly influential approach has been that of Norman (1986), further developed by Fischer (1991). Norman sees effective system design in terms of the mental models of the designer and the user: "The problem is to design the system so that, first, it follows a consistent, coherent conceptualization – a design model – and, second, so that the user can develop a mental model of that system – a user model – consistent with the design model." (Norman 1986, p.46). He then illustrates this with a diagram of their interaction:

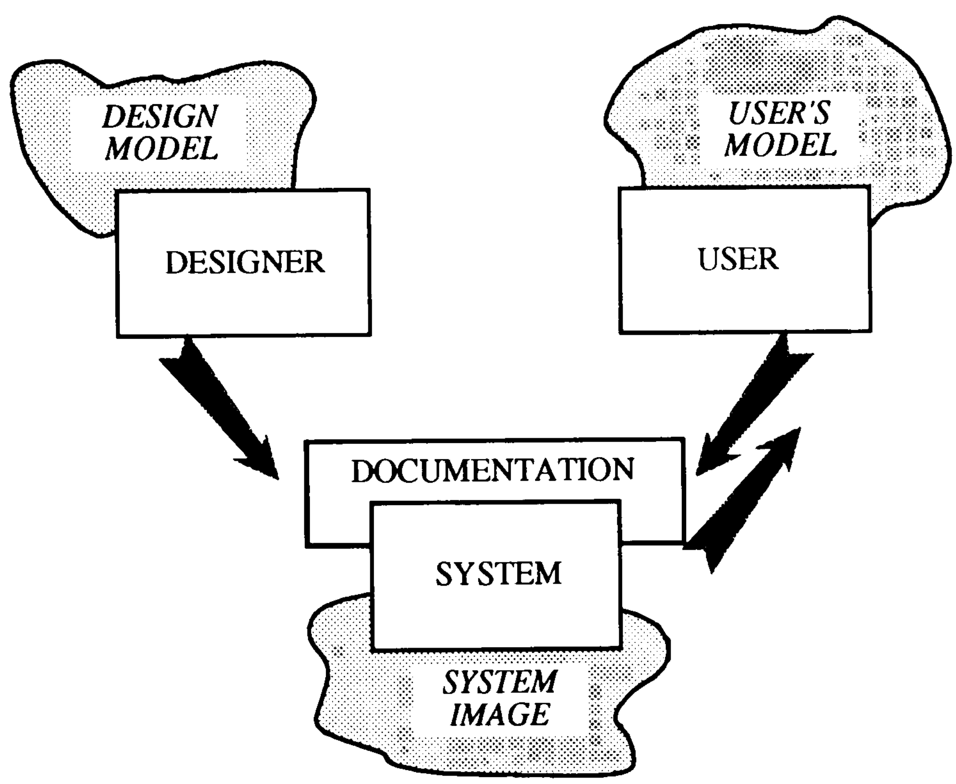


Figure 3.2: The design model and the user's model
(based on Norman 1986, p.46).

Building on Norman's terminology, Fischer (1991) breaks down the user's model into three components, shown as D_1, D_2, and D_3, in the following diagram:

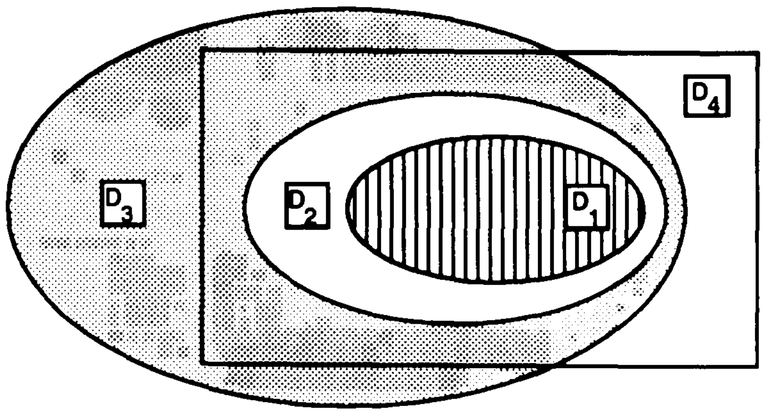


Figure 3.3: Levels of system usage
(based on Fischer 1991, p7).

The different areas correspond to the following:

- D_1, the subset of concepts the user knows and uses.
- D_2, the subset of concepts used only occasionally and not initially known.
- D_3, the user's model of the system (i.e. the set of concepts the user *thinks* exists in the system).
- D_4, the actual system.

3.4.3 Mental models and metaphor

The important distinction introduced by Fischer is between the set of concepts the user knows, D_1, and the set the user *thinks* exists, D_3. When a user is first introduced to a system, only D_4, the actual system, and D_3 will be present. When the system is based on a metaphor, Hammond and Allinson (1987) and Rogers et al (1988) suggest that the user's initial mental model, D_3, can be expected to correspond to the user's mental model of the metaphor vehicle.

The effectiveness of the metaphor can then be considered by comparing the features of the metaphor vehicle with the features of the system, the assumption being that a closer fit will mean a more effective metaphor.

This can then be expressed by considering the features of the vehicle and tenor as intersecting sets:

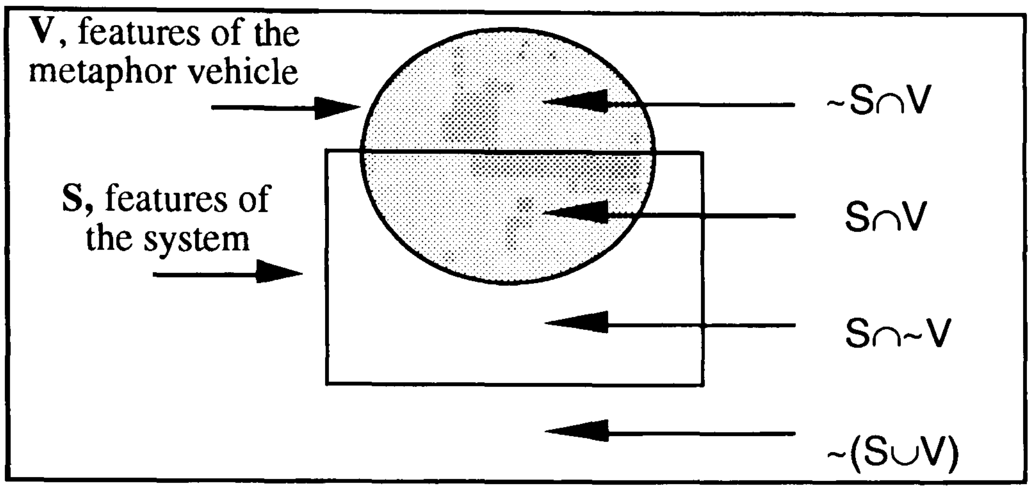


Figure 3.4: A model of metaphor at the human-computer interface (based on MITS 1994, p.11).

The first set represents the features of the vehicle, **V**, whilst the second represents the features of the tenor, in this case the computer system, **S**. This conceptualisation can be illustrated using examples taken from the Macintosh wastebasket:

~S∩V An example of a real-world feature not present in the Macintosh system might be the ability to upturn a wastebasket and sit on it.

S∩V This category includes being able to move documents from the desktop to the bin and retrieve them by taking them out of the bin, providing it has not been emptied.

S∩~V This includes the ability to eject disks by dragging them to the wastebasket.

~(S∪V) Although these features are neither part of the system nor the vehicle, consideration of features that *could* be appropriate to the system will be important when making choices about what to use as a vehicle and what functionality to include. For example, users could be allowed to open documents in the wastebasket without taking them out (although this would not be possible in the real world).

It should be noted that, in practice, the set **V** should not be taken as the *actual* features of the vehicle but the set of what the user *thinks* are the features of the vehicle – the user's mental model of the vehicle – as this is what the user's behaviour will be based on. Based on these definitions, Anderson et al (1994) suggest a number of inter-related factors that determine the effectiveness of a metaphor, depending on the degree of overlap between the two sets of features. The most important of these is that users should make appropriate inferences about the functionality of the system from their understanding of the vehicle (Douglas 1983; Carroll 1988; Smyth 1993). Anderson et al propose that problems will occur whenever there is a high proportion of $\sim S \cap V$ features compared to $S \cap V$ features. They describe this as 'conceptual baggage' brought to the interface by the vehicle – the features of a vehicle that are not utilised in a particular vehicle-system pairing.

This problem can be reduced either by selecting a vehicle with a restricted scope or by expanding the functionality of the system to include more features of the vehicle. However, as noted in Chapter 2, authors such as Johnson (1987) and Dieberger (1994a) have shown that blindly 'following the metaphor' can lead to restrictive systems in which the ability to do anything truly new is lost.

3.4.4 Testing the mental model approach

Anderson et al (1994; Smyth 1995b) carried out an experiment to look at the influence of metaphor choice on the user's mental model of the system. Three metaphor-based interfaces were developed, each supporting identical system functionality. The original system, 'Doorways', later shortened to 'Doors', was implemented at the Rank Xerox EuroPARC research centre, where it provided a test interface to the RAVE multimedia internal communication system (Buxton 1990; Gaver 1992). Office doors formed the main metaphor of the system, indicating the availability of another person. Three states of availability were supported: available (door open), unavailable (door closed) and 'busy' (door ajar). A 'busy' person could be 'glanced at' by a still video frame shown briefly to allow the user to decide whether to interrupt.

As reported in Chapter 2, metaphor is seen as "important because it provides us with a way of moving from known ideas and familiar concepts to new and unknown ones." (Cornell Way 1991, p.8). From the Star onwards, it has been assumed that interface metaphors should therefore be chosen from the existing working environment. To test this assumption, two metaphors were chosen which were familiar to users but did not form part of the office environment. 'Dogs' was chosen as a familiar concept irrelevant to office activity. 'Colours' was originally chosen as a metaphor-free indicator, simply using different colours to indicate the three states. However, the designers chose the three colours of traffic lights to emulate the three states, an implicit metaphor. Despite this, they were presented as simple blocks of colour and the underlying metaphor was not made more explicit. Each of the interfaces had a graphically presented equivalent for the three states of availability:

Table 3.1: States of availability.

State	**Doors**	**Dogs**	**Colours** (Traffic lights)
Available	Open	Dog standing	Green (Go)
Busy	Part-open	Burying a bone	Orange (Caution)
Unavailable	Closed	Asleep in basket	Red (Stop)

Subjects were presented with a task scenario which required them to act as a member of a small software development team. This task entailed attempting to contact various people and having to deal with each of the three states. The roles of other members of the software team were played by experimental stooges. All subjects completed a questionnaire to test how well they had understood the system, each statement being assigned to one of the sets outlined above.

As predicted by the experimenters' model, the greatest variation in the number correct responses occurred with the ~S∩V set, i.e. the 'conceptual baggage':

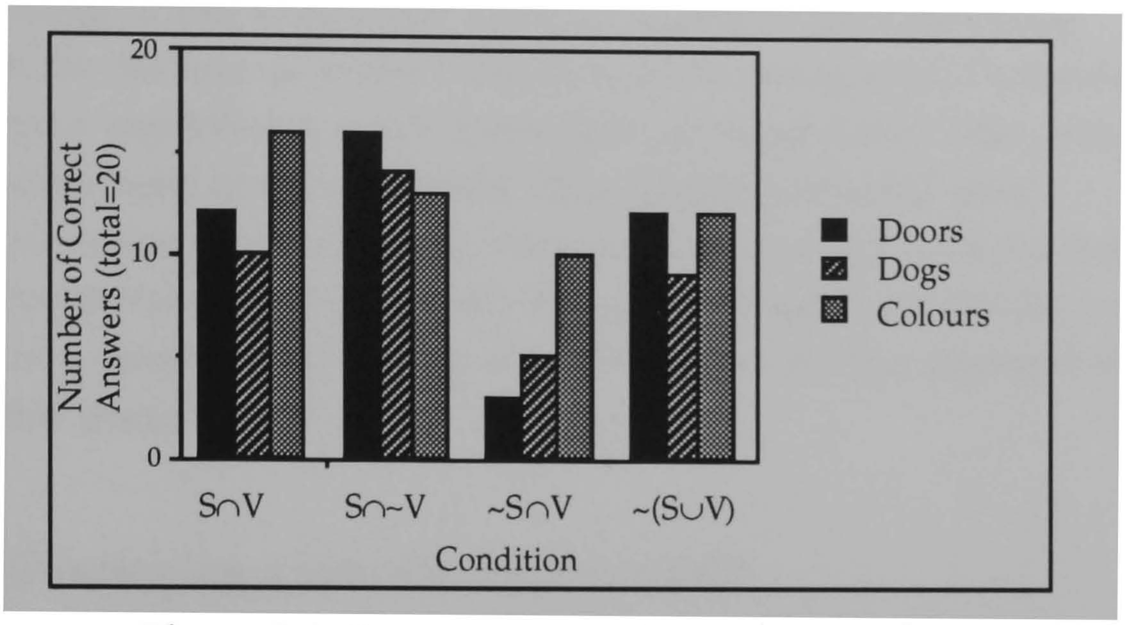


Figure 3.5: Correct answers for each metaphor
(based on Anderson 1994).

The Doors interface metaphor caused the subjects to make significantly more incorrect assumptions about the underlying system functionality, the number of correct answers in the ~S∩V condition being significantly less than in any other condition, or for any other metaphor. Despite this, subjects' confidence in their answers, which they also recorded, was as high as in the other sections. This misplaced confidence appeared to be due to the richness and contextual relevance of this vehicle, which had the effect of masking the boundary of the mapping between vehicle and system through a large amount of conceptual baggage. The effect of this baggage was exacerbated by the relative simplicity of the underlying system functionality.

In contrast to the Doors vehicle, it appeared that although Dogs provided a rich set of resources, these were largely inappropriate. Thus, while a degree of conceptual baggage was present, the lack of contextual relevance caused the effect to be reduced. In the case of Colours, the vehicle mapped only to a small part of the system functionality, apparently causing subjects to be aware of the boundary between the two. Despite this, subjects felt that the Doors interface was more intuitive, with far less of them asking for manuals than users of the Dogs and Colours interfaces.

In fact, no manual or other assistance was provided, even when requested.

The findings of this experiment certainly appear to provide strong support for the mental model view of interface metaphors. However, the experiment supported a much lower level of functionality than most applications used in the real world. Also, the subjects used were computer science post-graduates, who could be expected to be far more computer-literate than most users. Whether this approach would be useful in a more realistic setting will therefore be further explored further on in this thesis.

3.5 Developing a semiotic model of HCI

3.5.1 Other applications of semiotics to HCI

Other attempts have been made to apply semiotics to computing. The most expansive is Andersen (1990), but he offers a much lower level view than the conceptual level this thesis is dealing with. For example, Andersen devotes less than five pages (from over 400) to metaphor (Andersen 1990, p.155-159), and none to other tropes. Also, much of his discussion of semiotics is concerned with the phoneme or its visual equivalent, the grapheme, which de Saussure (1974, p. 4; 18; 66) explicitly excluded from semiotic relevance.

The most relevant semiotic approach to user interface design is that developed by de Souza (1993) and extended by Prates et al (1997). This depends on Eco's concept of 'unlimited semiosis' and has much in common with my own approach. I will examine their work in greater depth below, when I introduce the concept of unlimited semiosis.

3.5.2 The sign-function

The literature of semiotics is full of different models and personal terminology that can obscure understanding. Although de Saussure (1974, p.66-67) and Peirce (1985, p.5) came to similar conclusions in seeing the sign as inseparable from what it signifies, there are important differences in their models of this. Speaking of *parole* (spoken language), de Saussure draws a dual relationship:

> The linguistic sign unites, not a thing and a name, but a concept and a sound-image. The latter is not the material sound, a purely physical thing, but the psychological imprint of the sound, the impression that it makes on our senses. The sound-image is sensory, and if I happen to call it "material" it is only in that sense, and by way of opposing it to the other term of the association, the concept, which is generally more abstract....
>
> The linguistic sign is then a two-sided psychological entity that can be represented by the drawing:

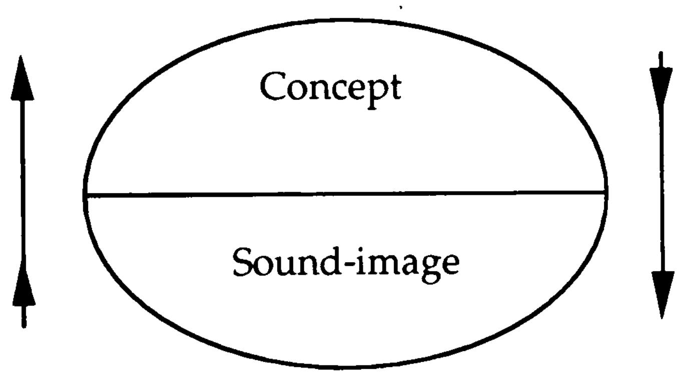


Figure 3.6: Concept and sound image
(de Saussure 1974, p.66).

He then introduces his favoured terminology for these:

> Ambiguity would disappear if the ... notions here were designated by three names, each suggesting and opposing the others. I propose to retain the word *sign* [*signe*] to designate the whole and to replace *concept* and *sound-image* respectively by *signified* [*signifé*] and *signifier* [*signifiant*]; the last two terms have the advantage of indicating the opposition that separates them from each other and from the whole of which they are parts. (de Saussure 1974, p.67).

Placing these terms in the diagram, we get the following, with the arrows standing for "the opposition that separates them from each other" :

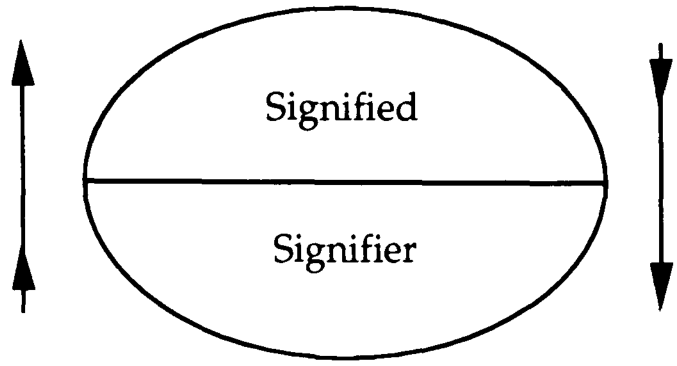


Figure 3.7: Signifier and signified.

Although this applies directly to spoken language, de Saussure is clear that linguistics forms only a part of 'semiology', and the signifier could be replaced by equivalent mental images for other types of sign. It is important to note that for de Saussure the sign is "a two-sided psychological entity"; the physical entity – the sound or the image – is not included as part of the sign.

In contrast to de Saussure's two part model, Peirce proposed a more complex model:

> A sign, or *representamen*, is something which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign. That sign which it creates I call the *interpretant* of the first sign. The sign stands for something, its *object*. It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the *ground* of the representamen.
> (Peirce 1985, p.5).

Comparing the two models, the following equivalencies appear to have been made:

Table 3.2: Comparing de Saussure and Peirce's concepts.

Concept	de Saussure	Peirce
The physical sign	–	Representamen
Perception of the sign	Signifier	Interpretant
What it signifies	Signified	Object

Note that Peirce's term 'ground' is similar to, but not the same as, the 'ground' of a metaphor defined by Richards in Chapter 2. After its introduction, Peirce rarely refers to the ground, preferring to keep to a triadic model: the representamen, the interpretant and the object. In this context it is necessary to add a cautionary note to any assessment of Peirce's work. As Gardner (1992, p.65-66) explains, Peirce tried to express his mathematical and philosophical ideas in terms of triads, seeing this as a fundamental form, and it is possible that he simply down-played the importance of the ground as an unwelcome fourth component.

Barthes adapted de Saussure to also produce a triad, though this does not equate to Peirce's. Barthes created his three-part model by the addition of the 'sign' as a concept in addition to, rather than simply combining, the signifier and the signified:

> [Any] semiology postulates a relation between two terms, a signifier and a signified. This relation concerns objects which belong to different categories, and this is why it is not one of equality but of equivalence. We must here be on our guard for despite common parlance which simply says that the signifier *expresses* the signified, we are dealing, in any semiological system, not with two, but with three different terms. For what we grasp is not at all one term after the other, but the correlation which unites them: there are, therefore, the signifier, the signified and the sign, which is the associative total of the first two terms. (Barthes 1973, p.112-13).

The "correlation which unites" the signifier and signified is not Peirce's representamen, the physical sign, although it may be closer to Peirce's term 'ground'. Unfortunately, the term 'sign' has now been used to refer to the physical sign (Peirce), the combination of two psychological concepts (de Saussure) and the correlation between these two concepts (Barthes). To avoid too much confusion, I will use *sign* in the commonest sense, the physical sign, but it should always be read as being in the context of the entire process. I will continue to use de Saussure's terms for the perception and cognition of the sign: the *signifier* and the *signified*, respectively (de Saussure 1974, p.67). For the correlation that unites these, in common with most semioticians, I use the term *signification*, as defined by Eco (1979, p.8). Eco draws a clear distinction between communication and signification:

> So let us define a communicative process as the passage of a signal (not necessarily a sign) from a source (through a transmitter, along a channel) to a destination. In a machine-to-machine process the signal has no power to signify in so far as it may determine the destination *sub specie stimuli*. In this case we have no signification, but we do have the passage of some information.
>
> When the destination is a human being, or 'addressee' (it is not necessary that the source or the transmitter be human, provided that they emit the signal following a system of rules known by the human addressee), we are on the contrary witnessing a process of signification – provided that the signal is not merely a stimulus but arouses an interpretive response in the addressee. This process is made possible by the existence of a code. (Eco 1979, p.8).

The *code* that Eco refers to is the set of rules that allows the process of signification to take place. Consider, for example, the handshake. We normally interpret a handshake according to a relatively simple code in which a firm handshake signifies a certain type of friendliness and trustworthiness. A Mason, however, has a quite separate code with which to interpret the same handshake, recognising certain arrangements of the digits to indicate common fellowship in the Masons. The two codes co-exist and both are culturally dependent: Japanese businessmen have to be taught the standard code of the handshake just as Westerners need to be taught the Japanese codes for bowing.

3.5.3 Classes of signs

Peirce (1985, p.8) identified three classes of signs:

- **Iconic**: a sign which resembles the signified (e.g. a portrait, a photograph, an x-ray, a diagram, a map).
- **Symbolic**: a sign which does not resemble the signified but which is purely conventional (e.g. the word 'stop', a red traffic light, or a number such as '2').
- **Indexical**: a sign which is inherently connected in some way (existentially or causally) to the signified (e.g. smoke signifies fire; thermometer, weathercock, knock on door).

Note that the use of the term 'iconic' does not correspond to the wide use of the term in relation to user interfaces. A named icon in a GUI, or the name of program or file in DOS, is a very special type of sign that I will refer to as an *interface sign*. An interface sign functions in many ways and could actually be placed in any of Peirce's three classes. In the simplest sense, the choice of characters by which we choose to name a file is entirely arbitrary: a **symbolic** sign. The most important power of the interface sign, however, comes from it being a particular form of **indexical** sign. Barthes comes closest to expressing this in his example of a woodcutter:

> Here we must go back to the distinction between language-object and metalanguage. If I am a woodcutter and I am led to name the tree I am felling, whatever the form of my sentence, I 'speak the tree', I do not speak about it. This means that my language is operational, transitively linked to its object; between the tree and myself, there is nothing but my labour, that is to say an action. This is political language: it represents nature for me only inasmuch as I am going to transform it, it is a language thanks to which I '*act the object*'; the tree is not an image for me, it is simply the meaning of my action. But if I am not a woodcutter, I can no longer 'speak the tree', I can only speak about it,

> on it.... I no longer have anything but an intransitive relationship with the tree; this tree is no longer the meaning of reality as a human action, it is an *image-at-one's-disposal..* (Barthes 1973, p.145-46).

For an interface sign, the signifier is existentially linked to its signified; a file is created by naming it; manipulating the interface sign manipulates the file it signifies (or an alias pointing to that file); 'deleting' the sign deletes the file or alias; neither can exist without the other. The language of our interaction with the computer is thus what Barthes terms 'political', 'speaking' the computer, 'acting the object,' with a direct impact on what the computer does, as distinct from, say, this thesis which speaks *about* the computer.

Much of the semiotic code with which we interact with the computer is concerned with this indexical nature of the interface sign. For example, it is possible to 'double-click' on icons in the Macintosh interface to open the document with its associated application. The action of double-clicking is unrelated to the desktop metaphor; it is unrelated to the interface sign itself; it is unrelated to the contents of the file; it is only related to the performance of the interface sign as an object in itself and its indexical link to the file.

Finally, there are some respects in which the interface sign is **iconic**, whether it is an actual icon in a GUI or a file name in a command language. Although an iconic sign 'resembles' its signified, this need not be a literal representation; it can be highly coded:

> Particularly deserving of notice are icons in which the likeness is aided by conventional rules. Thus, an algebraic formula is an icon, rendered such by the rules of communication, associations, and distribution of the symbols. ... a great distinguishing property of the icon is that by the direct observation of it other truths concerning its object can be determined than those which suffice to determine its construction. (Peirce 1985, p.11).

One 'likeness' of an interface sign lies in its file type, whether expressed by the characters at the end of the file name (TXT, EXE, etc.) or by the type of interface icon used. This is determined by the construction of the interface sign but, like the algebraic formula, it conforms to logical rules from which the user can determine other truths. For example, a file created with SimpleText will have an icon type determined by SimpleText but the user can deduce that the file can be opened by Microsoft Word. Other icons generated by graphics programs are iconic in the most trivial sense, in that they generate 'thumbnail' forms of the picture itself.

Before leaving this, I should point out that Eco has described Peirce's categories as "an untenable trichotomy" (Eco 1976, p. 178), saying that the terms are too vague to be useful and proposing his own far more complex typology (Eco 1976, p.218) with four aspects of a sign under which it can be classified in up to twelve different ways. While accepting that Eco's argument has value, Peirce's simple division is sufficient to draw attention to the particularly powerful indexical nature of interface signs.

3.5.4 Denotation and connotation

In the quotation above, concerning the woodcutter, Barthes considered the distinction between 'mythological' and 'political' speech or actions. This distinction deserves a brief explanation. As the quotation explains, 'political' language, or "language-object" changes things, whereas myth merely comments on it. Myth is *"is a second-order semiological system"* (Barthes 1973, p.114, italics in original) or, as he puts it in the quotation above, "a meta-language".

This distinction is not absolute. Every sign has both political and mythological elements, whether the text of which it is part is political or mythological. Every sign both 'speaks itself' and 'speaks about itself', two types of signification known as *denotation* and *connotation*. Consider the Apple logo as an example:

Figure 3.8: Apple logo.

At a simple level, this picture denotes an apple. But any picture of an apple could denote an apple. The combination of the rainbow colouring and the bite out of the side also says 'Apple Computer'. However, it has many connotations beyond this. Apple chose a picture, not a logo based on their name like IBM, DEC or most other competitors at the time. It is also coloured like a rainbow, a concept associated in America with California's ethnic mix of white, black, Hispanic and Asian and with the 'Rainbow Alliance': the name given to the loose cooperation between black activists, radical gays (pink) and the Greens.

The example of the Apple logo shows that a simple sign, created as a sign, can have many different significations among which it is not always easy to identify the denotation. Where a sign has not been created as a sign but simply exists as a sign, such as real apple, the distinction between denotation and connotation is easier. A real apple will have many connotations, heavily dependent on context. In a church it may become a symbol of the harvest festival or, in a slightly different context, of Original Sin. In a basket at the dentist or doctor it becomes a symbol of healthy eating or looking after one's teeth. However, an apple still maintains one simple denotation, that of an apple, whatever language we speak: an apple, *pomme*, or *apfel*.

Fiske claims that, "it is often easy to read connotative values as denotative facts; one of the main aims of semiotic analysis is to provide us with the analytical method and the frame of mind to guard against this sort of misreading" (Fiske 1982, p.92). According to this view, it might be argued that there is a single 'literal' denotation of the Apple logo – the picture denotes an apple, just as a physical apple itself denotes an apple – with all other meanings representing cultural connotations of the sign. It is possible to argue with this interpretation in that the denotation itself is not without cultural or ideological connotations; in another part of the world, the picture might be interpreted as a quite different fruit, perhaps an inedible one. It could therefore be argued that the true denotation of the Apple Computer logo is 'Apple Computer' as this is the most unambiguous interpretation of this specific portrayal of an apple.

In contrast to Fiske, Hall claims that there are dangers in seeing denotation and connotation as distinctly separate:

> The term 'denotation' is widely equated with the literal meaning of a sign: because the literal meaning is almost universally recognised, especially when visual discourse is being employed, 'denotation' has often been confused with a literal transcription of 'reality' in language - and thus with a 'natural sign', one produced without the intervention of a code. 'Connotation', on the other hand, is employed simply to refer to less fixed and therefore more conventionalised and changeable, associative meanings, which clearly vary from instance to instance and therefore must depend on the intervention of codes...
>
> But analytical distinctions must not be confused with distinctions in the real world. There will be very few instances in which signs organised in a discourse signify only their 'literal' (that is, nearly-universal) meaning. In actual discourse most signs will combine both the denotative and the connotative aspects (as redefined above)...
>
> The terms 'denotation' and 'connotation', then, are merely useful analytic tools for distinguishing, in particular contexts, between not the presence/absence of ideology in language but the different levels at which ideologies and discourses intersect.
>
> (Hall 1980, p.132-33)

Hall points out that ideologies and discourses can intersect at different levels. I will pursue this with a recursive model in which denotation is merely a 'seed', the first of the many layers of signification recursively formed.

3.5.5 Layers of signification

Returning to Barthes' model of the sign, he illustrates the distinction between political language (denotation) and the meta-language of myth (connotation) in a simple diagram (Barthes 1973, p.115). I have reproduced this below, adapted to fit the terminology I am using:

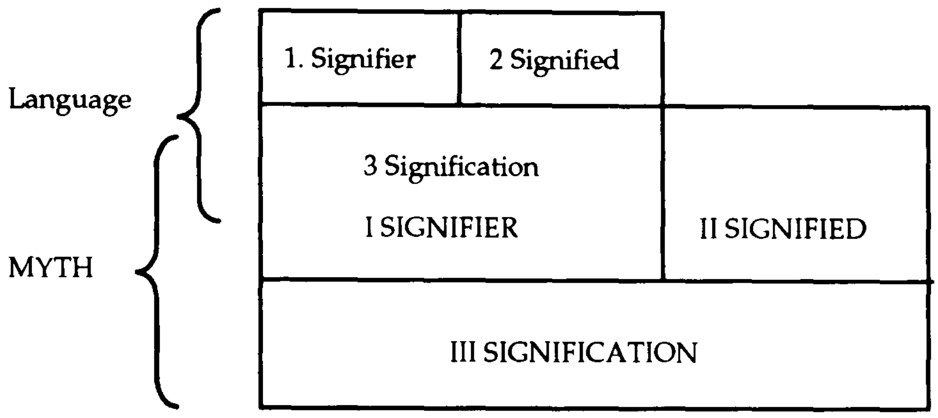


Figure 3.9: Signification as a signifier
(Adapted from Barthes 1973, p.115).

Or, as put by Chandler (1995):

> In semiotics there are different 'orders of signification' (levels of meaning). Semioticians distinguish (perhaps sometimes too tidily) between denotation - what a sign stands for - and connotation - its cultural associations. References to the signifier and the signified are sometimes described as the first order of signification - that of denotation, whilst connotation is described as a second-order signifying system.
>
> In conventional semiotic terms, connotation uses the first sign (signifier and signified) as its signifier and attaches to it an additional signified. Connotations 'derive not from the sign itself, but from the way the society uses and values both the signifier and the signified.' (Chandler 1995, p.1).

We can see from his diagram that Barthes' concept of myth is as a second-order signifying system, taking the total sign-function and treating it as the signifier within a higher level sign-function. If we accept that this is possible, is there any reason not to continue the recursion indefinitely?

When Peirce proposed his triadic model of the sign, he appeared to see it as part of an unlimited recursive model of signification:

> A *Sign*, or *Representamen*, is a First which stands in such genuine triadic relationship to a Second, called its *Object*, as to be capable of determining a Third, called its *Interpretant*, to assume the same triadic relation to its Object in which it stands itself to the same Object... [The Third] must be capable of determining a Third of its own; but besides that, it must have a second triadic relation in which the Representamen, or rather the relation thereof to its Object, shall be its own (the Third's) Object, and must be capable of determining a Third to this relation. All this must equally be true of the Third's Thirds and so on endlessly; and this, and more, is involved in the familiar idea of a sign; and as the term Representamen is here used, nothing more is implied. A Sign is a Representamen with a mental Interpretant. (Peirce 1985, p.6)

Peirce's writing style is somewhat opaque and this passage could be interpreted in a number of ways. I would suggest that Peirce sees the totality of the sign as a recursive process. "The Third... must be capable of determining a Third of its own" and "[the Third] must have a second triadic relation in which the Representamen, or rather the relation thereof to its Object, shall be its own (the Third's) Object, and must be capable of determining a Third to this relation." Placing this into the terminology I have been using:

1 The signification must determine a signification of itself

2 The relationship of the signifier to the signified also becomes a new signifier, forming a new signification.

Peirce's view does not stop here, "All this must equally be true of the Third's Thirds and so on endlessly; and this, and more, is involved in the familiar idea of a sign." In other words, the recursion is endless. Obviously, our minds have a finite capacity so that the recursion must remain finite, but in mathematical terms it is unbounded.

As an example, we can see destructive recursion build up in interpersonal relationships by reading higher signification into a sign. We might say, "He's being so nice, he must be after something." We distrust someone for being too 'nice', so they act more 'nicely' to overcome our distrust which increases, so they act even more 'nicely' and so on. An extended analysis of the manner in which these loops can occur can be found in Laing (1966) et al, while Laing also shows how this analysis can be used to examine specific examples (Laing 1970).

3.5.6 Semiotic engineering

Another semiotics based approach to user interface design exists in the form of 'semiotic engineering'. Originally formulated by de Souza (1993) for the design of user interface languages, the approach has also been applied to multi-user systems (Prates 1997) and the gathering of user requirements (Pimenta 1997). Eco splits his Theory of Semiotics into two complementary parts, a Theory of Codes (Eco 1979, p. 48-150), relating to signification and a Theory of Sign Production (Eco 1979, p.151-313), relating to communication. Semiotic engineering is based on his Theory of Sign Production (de Souza 1993, p.754), whereas this thesis is based on his Theory of Codes and is thus complementary.

Signification deals with what a sign signifies to the user, not what has been done to generate it. The existence of a sign does not necessarily imply that there has been any conscious design process, as when red spots signify measles to a doctor. This thesis examines the signification of the metaphor to the user, not the designer's intentions. As demonstrated by the following diagram, in which she shows the relationship between cognitive engineering and semiotic engineering, de Souza is concerned with the design process, an important concern but complementary to mine:

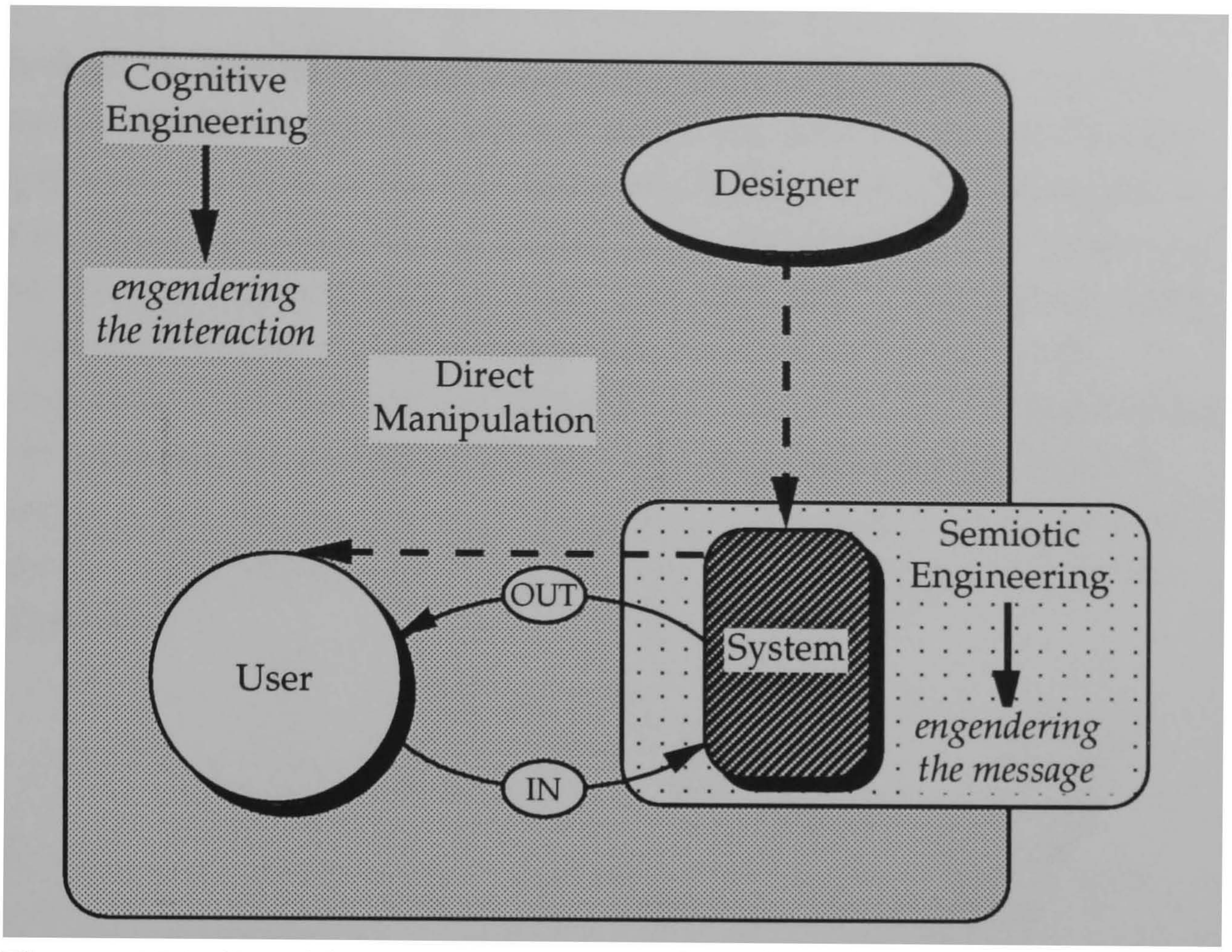


Figure 3.10: Cognitive and semiotic engineering (de Souza 1993, p.761).

In taking this approach, de Souza looks at Kammersgard's (1988) classification of HCI into four perspectives: the systems perspective whereby users are seen as data entry components; the dialogue-partner perspective, where users and systems are seen as equal partners in conversation; the tool perspective whereby systems are tools to be employed by users; and the media perspective in which systems are viewed as a communication medium through which people pass messages. De Souza concentrates her discussion on the dialogue-partner and media perspectives, claiming that systems are "message senders and receivers at the immediate interface level, but they are also achieved messages, themselves, sent from designers to users through the computational *medium*." (de Souza 1993, p.753). This thesis does not depend on this perspective, looking only at what the computer interface signifies to the user, regardless of the designer's intention when constructing it.

The work on semiotic engineering raises a concern about one of Eco's concepts which at first appears similar to the layers of signification. Eco's concept is of 'unlimited semiosis' (Eco 1979, p.71), which he further develops as 'infinite semantic recursivity' (Eco 1979, p.121). Put simply, it appears that Eco is pointing out that a sign can only be defined through other signs (or parts of the sign-function), leading to a need to define those signs, thus introducing further signs, *ad infinitum*. Eco points out that *"Semiosis explains itself by itself"* (Eco 1979, p.71, italics in original), in the same manner as a dictionary defines words with the words themselves. This does not appear to entail the same form of recursion as in the model I am proposing, but this may be a difference in personal interpretation. Certainly semiotic engineering appears to interpret Eco's concept in just this way, as this diagram, taken from Prates et al (1997, p.29) shows:

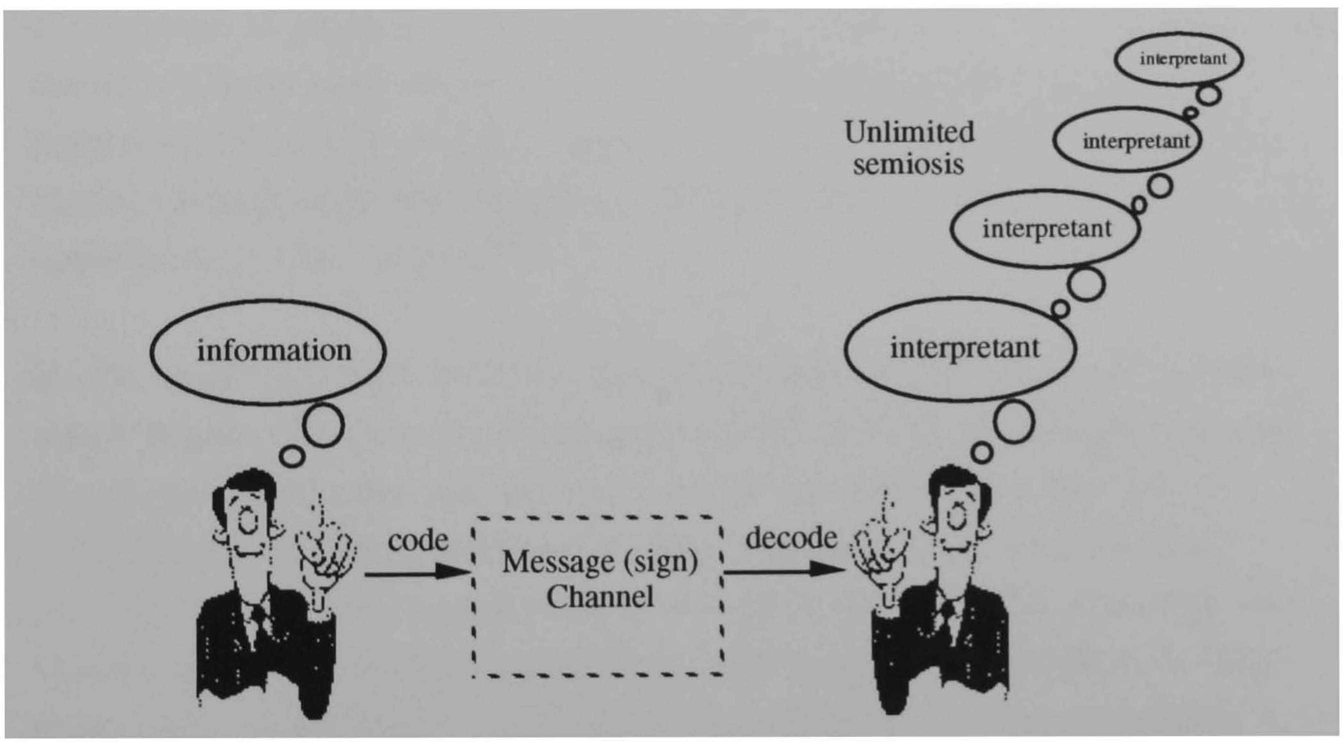


Figure 3.11: Communications process.

Whether this interpretation of unlimited semiosis is correct does not, in fact, affect this thesis. Having introduced the concept, Prates et al do not investigate it but turn their attention towards the ways in which multi-user systems mediate interpersonal communications. They do not consider the intercommunication between the computer and the person in semiotic terms, only the intercommunication between the people as mediated by the computer system and the interfaces. The concern of the authors is to use semiotics to examine the communications between users, helping designers to create interfaces that will support the multiple layers of signification that exist in natural communication. By contrast, this thesis examines the layers of signification generated by the intercommunication between the user and the computer.

3.5.7 The number of layers of signification

Peirce and Barthes' models of recursion suggest a mechanism by which multiple layers of signification could exist in any sign. The implication is that this signification will extend infinitely. If this is so, then the designer can never hope to be fully aware of the impact of the interface on the user. The problem is analogous to the more common question, "What is the longest sentence in the language". The answer to this question is that if one were to propose a 'longest sentence', and label it 'S', then one could create a longer sentence by saying "S is the longest sentence in the language," with the actual content of S substituted for the symbol. A more relevant question becomes, "What is the highest level of signification *that matters?"*

In the context of user interface design, I propose that this level will be much higher than the levels usually attended to in the design process. Computer companies are certainly aware of very high levels of signification in the promotion of their products. For example, the Macintosh was introduced with a television commercial, showing the Macintosh as a revolutionary device, able to smash a totalitarian 'Big Blue' state, while IBM's AS400 has been advertised with a picture of a slave breaking his chains. Messages like this may be beyond the immediate scope of the interface designer, but an interface is designed in the context of a company's advertising and signification at this level is certainly not beyond the scope of the advertiser. Even religious symbolism has not been ignored by computer manufacturers: Apple has placed much faith in its 'Evangelism' division (Kawasaki 1998).

It should not be surprising that the main commercial application of semiotics at these levels has been in advertising where the advertiser wishes signs to be associated with positive rather than negative metaphoric connotations. The use of semiotics in advertising and the analysis of advertising is widely discussed in various papers edited by Blonsky (1985), as is the use of semiotics in media such as television, cinema and political posters. The importance of semiotics in graphic design has also been recognised, with some attempts to apply these principles in computer interfaces such as that by Aaron Marcus (1983). More recently, industrial designers have been trying to adopt semiotic concepts, although they seem very divided as to what type of contribution semiotics can make: see Vihma (1989) for a wide range of examples. However, the central concern of this thesis is whether semiotics supplies effective tools for helping user interface designers, particularly in the use of interface metaphors.

3.5.8 Semiotic systems

The quotation from Weinrich in Chapter 2 referred to 'semiotic systems' but, as with other authors, offers no explicit definition of the term. For the purposes of this thesis, I therefore propose the following definition, based on the manner in which the term is commonly used in the literature:

- A semiotic system consists of two or more related signs.
- Each sign in a semiotic system is capable of changing the signification of other signs within the system.

As in the definition of a sign above, a semiotic system does not have to be intended as such. To take the example of spots signifying measles to a doctor, the presence of other signs such as a high temperature and a headache might change the signification to one of meningitis. Thus the symptoms of the body taken together form a semiotic system. As with other aspects of semiotics, the definition of 'related signs' is dependent on the observer, who 'reads' meaning into the signs: a practitioner of alternative medicine might see the positions of the planets as part of the same semiotic system as the spots in coming to a diagnosis. It is also important to note that other signs within the system need not be physically present to affect the signification. In the standard traffic light sequence, red and amber signify that the green light is about to come on, whereas an amber light alone signifies that the lights are about to turn to red. The lack of a red light in the latter case is as much of a sign as the presence of the red light in the former, adapting the signification of the amber light accordingly.

3.6 Applying semiotics to HCI

3.6.1 Questions and assertions

Some of the issues raised in Chapters 2 and 3 depend on underlying assumptions; others imply consequences that need to be tested. In this section, I will draw these out as explicit assertions and questions and introduce the analyses, experiments and studies that I propose to use to establish the validity of my approach. The subsequent chapters will then describe these tests in more detail, showing the results and conclusions that can be drawn from them.

3.6.2 Tropes in HCI: a proposed analysis

The first assumption, raised in Chapter 2, concerns the complexity of user interfaces as semiotic systems:

Assertion 1	A user interface is a sufficiently complex semiotic system (even if not a true language) to develop through metaphor and metonymy, as natural languages do.

This assertion is relatively easy to check, by looking at the degree to which an existing interface makes use of tropes, particularly metaphor and metonymy. As recent interfaces have been explicitly based around metaphor, my analysis was carried out on an older version of the MS-DOS command language.

3.6.3 Semiotic analysis of user interfaces: a proposal

The descriptions of the semiotic sign-function imply a multiplicity of signification, even in the simple signs that make up a user interface.

Assertion 2	Layers of signification are so numerous that it must be quite easy to uncover many of them in any interface.

Before devising a more structured approach to semiotic analysis, I carried out an ad hoc analysis of the Macintosh user interface. Starting with a small part of the interface, I looked at what it appeared to signify in as many ways as possible, what these significations implied, and so on. This is highly subjective but demonstrates that such an analysis can be carried out.

3.6.4 Comparing mental models and semiotics

Although the work in cognitive psychology described in Sections 3.4.3 and 3.4.4 appears to confirm the usefulness of an approach based on the user's mental model of the metaphor vehicle, it raises a number of important questions:

Question 1	Does it matter whether users form accurate models of the system?
Question 2	Would 'real world' users behave differently?
Question 3	Are the results valid for more complex computer systems and interfaces?

These questions are answerable by echoing the original experiment with a more realistically complex system and subjects taken from the likely user group for the system. Performance at the tasks set can then be compared with the apparent accuracy of the user models of the system.

3.6.5 Layers of signification

It has been suggested that, by bringing in wider aspects such as social factors, semiotics offers a richer view of metaphor in HCI than is provided by other approaches. This also suggests the possibility that the choice of metaphor might influence the levels of signification that users naturally adopt:

Assertion 3	Interface metaphors create many different forms of signification not accounted for by the mental model approach.
Question 4	Does the type of metaphor affect the forms and levels of signification?

The simple classification of interface metaphors types given in Section 2.4.4 was used to choose three metaphors that might be expected to carry very different signification. These were then used as a basis for the experiment. Signification was looked at by examining the open-ended questions in the questionnaire completed by the subjects, looking at the terms they used when describing the system. The results, and the complete experiment, are described in Chapter 4.

3.6.6 Using semiotics to support interface design

Earlier in this chapter, a recursive model of signification was put forward that forms the basis for practical application of semiotics to user interface design. This is because the act of signification itself forms a sign: the fact that a signifier is associated with a specific signified by a particular person is significant in itself:

Assertion 4	The recursive nature of signification leads to a structured model of multiple layers of signification.

A simple mechanism is suggested to test this:

Assertion 5	Further layers of signification can be uncovered by asking of each layer, "What does that signify?" or, more simply, "What for?"

The simplest way to test this assertion is by interviewing users of computer interfaces, continually asking, "What is that for?" in response to their answers. This will also lead to the answer to a question posed earlier in this chapter:

Question 5	What is the highest level of signification that matters to the user?

It is suggested that this technique could be used by interface designers to discover aspects of the user interface they might otherwise have overlooked. The interviews with users are described in Chapters 5 and 6.

4 Preliminary studies

4.1 Introduction

Many of the assertions and questions introduced in Section 3.6 must be dealt with before more intensive research can proceed. These preliminary studies will also demonstrate some of the potential of the methods used to investigate them. Each preliminary study is based on a particular approach: rhetorical analysis, semiotic analysis, laboratory experiment and interview. The results of these studies are given below.

4.2 Analysis by trope of MS-DOS

4.2.1 Method

The first assertion put forward in the previous chapter was that a user interface is a sufficiently powerful semiotic system (even if not a true language) to naturally tend to develop through metaphor and metonymy. Computer literature shows few references to metaphor before the early 1980's, when the previously cited work by Carroll et al and the Xerox team started to appear. Terms used in MS-DOS (Microsoft Disk Operating System) have almost all been taken from previous command languages such as UNIX shell commands or CP/M and therefore pre-date any strong conscious decisions to employ metaphor. Consequently, it is reasonable to suppose that, where metaphor has been used in this language, it has not been introduced deliberately. In other words, the MS-DOS vocabulary provides a good test of any 'natural' tendency for metaphor and metonymy to play a major role in the evolution of user interfaces.

MS-Windows has adopted many of the metaphor-based features of the Macintosh interface and this might be expected to influence the terms used in more recent versions of MS-DOS. I have therefore analysed the commands listed in a manual for MS-DOS version 3.3 in 1987 (Compaq 1987). To carry out this analysis, I have consulted the definitions of the terms given in the complete Random House dictionary from the same year (Flexner 1987).

Many English words show elements of metaphor and/or metonymy in their evolution. For example, the word 'type' comes from the Greek τψποσ (typos), the act of striking or making a mark, or the stamp used to make that mark. This has been applied, literally, to movable type used for printing and then to the use of a typewriter. The use of the verb 'type' when using a wordprocessor is a simple metaphor which is now long dead. The MS-DOS command 'TYPE', however, has made a further metaphoric translation in its adoption by the MS-DOS command language. The purpose of the MS-DOS command is defined as 'to display the contents of a file' on the screen (Compaq 1987). This is obviously not the literal meaning of 'type', even in its newest form, but a new metaphor meaning, "Put the contents of this file up on the screen *as if* someone were typing it at the keyboard."

Some computer terms have already become assimilated into the language. For example, the words 'disk' and 'program' (particularly with U.S. spelling) have assimilated into the language, even though their use in computing was originally metonymic and metaphoric respectively. As MS-DOS was primarily developed in the USA, I used an American dictionary from the same year to arbitrate on whether the metaphor or metonymy had become assimilated. Where the dictionary lists the word as a computer term and the word is used in that sense in MS-DOS, I have therefore listed it as a dead metaphor or metonymy.

4.2.2 Results

The results are shown in the table below:

Table 4.1: Trope analysis of MS-DOS.

Commands	Metaphor	Metonymy	Literal	Other
	19 live	13 live	22	6
	12 dead	5 dead		
72	31	17	22	6

The total number of commands is less than the sum of the other categories. This is due to some commands, such as 'FASTOPEN', appearing in more than one category. In this case, the command copies file and directory locations into memory, in order to allow faster opening of them. Thus the command is named after a feature that is not the principal action of loading information into memory, but a feature resulting from that action (metonymy), that of allowing a file to be 'opened' (dead metaphor) faster. Some commands combine metaphor or metonymy with literal elements. In such cases, the literal element is ignored, as understanding of the command depends on the user's understanding of the metaphor or metonymy.

Commands listed as metonymy are of two types. In a command such as 'DISKCOPY', the term 'disk' is a metonym (though now assimilated into the language) for naming a particular type of data storage device. In most examples, it is the command itself which has a metonymic name. For example, the 'REPLACE' command can be used to replace the contents of one file with those of another but is equally likely to be used to add new information to an existing directory *without* replacing any existing files – in practical terms a diametrically opposed function. Thus the term 'REPLACE' is only one of two equally important attributes of the command.

In addition to these examples, limited metonymy (not shown in the table) is common. For example, the 'CHDIR' command is used for changing the current directory but, used with no parameters, can also provide information about the current directory. As it is named after only one of its uses, this could be classed as metonymy but changing directory is its major use and the name is predominantly a literal description of the command.

The category 'other' has been included to cover commands derived in other ways or of uncertain origin. Many are 'architectural' terms whose meaning was specific to the computer architecture at the time. Current use is more likely to be metaphoric. For example, most MS-DOS programs now run within windows and a command such as 'CLS' (clear screen) will not act on the entire screen. It will clear a window *as if* that window were a screen. However, the command was not created as a metaphor and is thus not included as a metaphor.

4.2.3 Conclusions

The original assertion to be tested was:

Assertion 1	A user interface is a sufficiently complex semiotic system (even if not a true language) to develop through metaphor and metonymy, as natural languages do.

The analysis found metaphor and metonymy present in the majority of MS-DOS commands. Only 22 of the 72 commands were purely literal, with 48 derived from metaphor and metonymy. By 1987, seventeen of these had already been assimilated into the language, giving some indication of the speed at which this process takes place.

Unlike the Macintosh, the use of metaphor in MS-DOS is not explicit and thus not structured around an underlying concept, such as the desktop. Literal expressions are used where a suitable context-free word or abstract noun has been identified, such as 'copy', but other adoptions have been in a haphazard manner. This analysis does not, of course, prove that a user interface is a semiotic system of a level of complexity comparable to a natural language, only that it is sufficiently complex to be capable of accumulating metaphoric and metonymic expressions.

4.3 A semiotic analysis of the Macintosh interface

4.3.1 Method

Barthes (1973b) and Eco (1987) have used the principles of semiotics to analyse various signs common in our society. Barthes and Eco do not offer methods for their analysis, although others have suggested procedures, such as Chandler (1995). An analysis of this type is highly subjective and demands considerable skill if it is to be complete. I do not suggest that it is practical or desirable for all designers to become experts in semiotic analysis. However, it is worth considering whether this sort of semiotic analysis can be applied to computer interfaces. If it is easy to uncover many layers of *potential* signification, this is sufficient to demonstrate the potential for such techniques in the analysis of user interfaces.

This exercise tests Assertion 2, that there is a massive range of signification inherent in a sign-system such as a user interface. If this is the case, it would be impossible to uncover every signification generated by the interface but the assertion may be regarded as valid simply by uncovering potentially useful significations which might not otherwise be noticed.

4.3.2 Analysis

The method employed was to start with a small part of the Macintosh user interface, showing a dialogue in progress as an example:

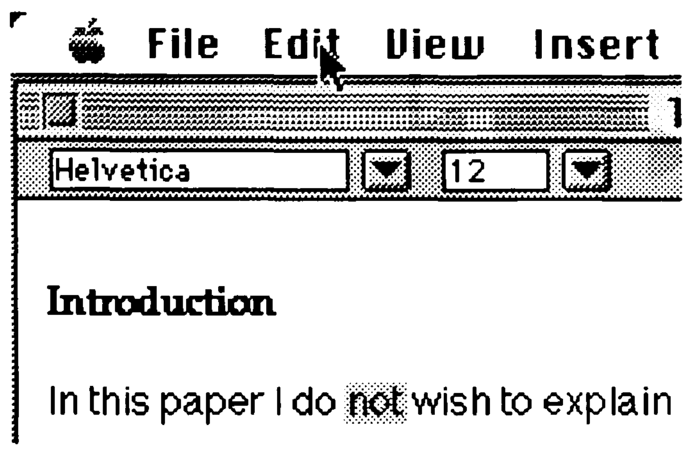


Figure 4.1: Part of a Macintosh interface.

Semiotic analysis is a self-reflective technique in which the researcher repeatedly asks him or herself, "What does this signify?" Carrying this out myself I found that, for example, the words on the screen signify:

'This is a sentence in the English language.'

The vocabulary used in the sentence, its spelling and its syntax also imply:-

'This is being written by someone who has received a reasonable education in this language.'

Note that the truth of the statement is not relevant to this form of analysis. Someone might have laboriously constructed a sentence in an unfamiliar language in order to falsely signify this signification. This does not make it a non-signification, only a false signification, possibly one the writer intended when writing the sentence. Eco has pointed out that the ability to lie is at the very heart of semiotics.

> *Every time there is a possibility of lying, there is a sign-function* : which is to signify (and then to communicate) something to which no real state of things corresponds. A theory of codes must study everything that can be used in order to lie. The possibility of lying is

> the *proprium* of semiosis...
> (Eco 1976, p.58-59 italics in original)

The potential of an interface sign to be misunderstood by the user is certainly of importance to the designer and must be allowed for in any analysis of its signification. In the case of my analysis, the statements are self-justifying in that I honestly believed that the sign could signify the levels I drew out in the analysis. Whether the sign was intended to carry this signification or it resulted from my mis-interpretation, other users could also make the same mistakes. It would be possible to draw a false analysis only by lying to oneself. If I claimed that the sign signified that 'the sun is hot', the statement would be dishonest, even though the sun is hot. There was nothing in the interface sign presented above to lead me to that signification, nor is there likely to be for any other analyst.

Continuing with this self-questioning technique, I uncovered 26 layers of signification. These started with simple statements, such as the fact that the interface is presented in the English language, or that it is a graphical interface. As the implications of these were considered, higher levels of signification were uncovered, such as the manner in which the Macintosh signifies 'this is a Macintosh' (not a PC), which in turn led to considerations of the relative images of the two architectures, the attitudes of their users towards them and even the potential political statements implied by them. The full list of significations uncovered by this exercise is given in Appendix B.

4.3.3 Conclusions

The analysis presents sufficient evidence that the original assertion was valid:

Assertion 2	Layers of signification are so numerous that it must be quite easy to uncover many of them in any interface.

It might be thought that the higher levels of signification are far removed from the Macintosh interface, but consider the advertising for the Macintosh. In 1997, advertisements were linked to the film 'Independence Day', in which the Earth is saved from an alien invasion by a scientist using a Macintosh. The slogan at the end was "Apple: The Power to Save the World." In a simpler way, the interface is 'selling' itself and the system to the user. This may be through a fiction, such as the Macintosh advertisement, or even through dishonesty – a false signification. One example of such dishonesty was examined in Chapter 2 (see figure 2.5), in which personification or user-friendliness can be used to signify a level of intelligence which the system does not possess.

Many of the individual points in this analysis are arguable, but it demonstrates that many layers of signification are likely to be present and even someone like me, who has never carried out a semiotic analysis before, can easily uncover many of them. Even the higher levels of signification have the potential to affect the ways in which users interact with systems and are thus factors that the designer might beneficially consider if the system is going to be used, and to be used effectively.

4.4 Comparing metaphor categories

In the previous chapter, an approach based on mental models was considered as the most promising alternative to a semiotic approach in investigating this field. In particular, an experiment by Anderson et al. (1994) was described in which they attempted to look at the match between the user's mental model of a system and the actual system functionality. This was based on the assumptions that the user forms a mental model of the system, and that the accuracy of the mental model affects usability. Their results showed that the greatest variation in accuracy between the interfaces occurred when examining concepts inherent in the vehicle but not present in the system. They described this as 'conceptual baggage', implying that too much conceptual baggage would hamper usage of the system. This obviously depends on accepting the underlying assumptions about the formation and accuracy of mental models.

It is easy to imagine that users can form mental models of systems as simple as those used in the experiment, but it is more difficult to imagine users forming a mental model of a commercial application which is far more complex. Model formation may well also be easier for computing students such as those who took part in their experiment than for the population in general. The question therefore must be asked as to whether the ability to form an accurate mental model plays any part at all in usability. I therefore developed a new experiment to test this assumption. Although based on the previous experiment, mine was not a development of it or a companion to it but a questioning of its underlying basis.

The experiment was based on the three categories of interface metaphor introduced in Section 2.4.4. In order to compare the results with the experiment reported in Chapter 3, described in Anderson et al (1994), the basic experimental method was identical. The experiment took place at BIBA (Bremer Institut für Betriebstechnik und angewandte Arbeitswissenschaft an der Universität Bremen). It should be noted that the experiments took place in German but, with help from Stephan Keuneke of BIBA, I have translated the instructions and other material into English for this thesis.

Various people at BIBA involved in the MITS project (including Hans Panse, Matthias Jankowiak and Stephan Keuneke) had developed ideas for metaphor-based interfaces. Based on an independent assessment of correspondence to the three metaphor classes by Christian Heath and Paul Luff from the University of Surrey, I took the three considered to mostly closely correspond to the categories and used them as the basis for developing working prototypes, where appropriate, further developing each to closer correspondence to its relevant category. All metaphors were presented as direct manipulation graphical user interfaces:

Spatial	MILAN	A room-based metaphor
Interactional	Link-Journal	A publishing metaphor
Activity-based	Little People	An agent-based metaphor

The interfaces were not intended to be original and, indeed, represent three of the commonest metaphors used for CSCW systems, as mentioned in Chapter 2: rooms/offices, agents/guides and books/newspapers. Each interface was examined separately, presenting an identical scenario to the subjects carrying out the experiments. Care was taken to avoid the use of metaphor in the description of the task involved in the scenario.

4.5 The Three Systems

4.5.1 Functionality

Each of the systems had the same underlying functionality and the same communications protocols. The first of the systems to be built was MILAN (Multimedia Industrial LAN). An earlier version of MILAN is described in Condon (1990), and in Hämmäinen and Condon (1991) where the use of the room metaphor for real-time interaction is compared with that of the form metaphor for a non real-time CSCW system.

The following functions were available to the users in the experiment, though most were not required for the scenario:

- Audio-video communications, including multiview video and multiway audio.
- File handling facilities, including archiving, file transfer across the network and shared visibility of files on other machines.
- Email send and receive, including local address books, etc.
- Shared and private work spaces, with partitioning of shared work spaces by, for example, project or workgroup.
- Personal organisation facilities, such as calculator, notepad, and personal address book.
- Auto-logging and playback of video logs.
- Shared drawing facilities, including a wide range of drawing and text manipulation tools. Each user also has a labelled pointer, seen by the other users.

Some functions were removed from the interface as too heavily embedded in the spatial metaphors used by MILAN and therefore not implementable in the other interfaces. These included maps for high level orientation and virtual reality facilities to 'walk round' a three-dimensional building, etc. The system was built in SuperCard 2.0, an object-oriented prototype development environment, though not a complete object oriented language (Allegiant 1997). This allowed the creation of the two new systems, Little People and Link-Journal, with very different user interfaces but identical underlying functionality. As they used exactly the same objects and methods as MILAN, they also possessed the same response times, allowing comparison of the interfaces alone.

The scenario required the subject to set up a multiway audio/video conference, send an email and use the shared pointer. The additional functions available might give the users better information on context, helping the user to find the right functions; alternatively, the additional functionality and resulting interface complexity might confuse the user.

4.5.2 The Task

The users were given a scenario with a series of tasks to carry out concerning the design of a chocolate box. This was chosen to reflect the type of activity which takes place in engineering design but was deliberately set as a non-manufacturing task to avoid technical issues, such as arguing about which machine tool to use, that might get in the way of the experiment. No time limit was given but it was suggested to subjects that the experiment should not take more than 'around half an hour'. Although there is no evidence that, for example, spatial metaphors are more useful for spatial tasks, to eliminate possible bias the main tasks were chosen to cover the three types of task equally:

Set up an audio/video conference (interactional), involving:

- work out the audio-video controls,
- find the 'customer' (an experimental stooge) in the address book,
- set up a two-way audio-video link to the customer,
- discuss the changes over the audio-video link.

Use the shared drawing facility (spatial), involving:

- find the correct drawing,
- open this with the shared drawing facility,
- choose the pointer tool,
- use this to identify the changes to be made.

Mail a report to a colleague (activity-based), involving:

- find and open the email facility,
- find the correct email address for the 'colleague',
- compose a message about the changes made,
- send the message.

4.5.3 Interfaces

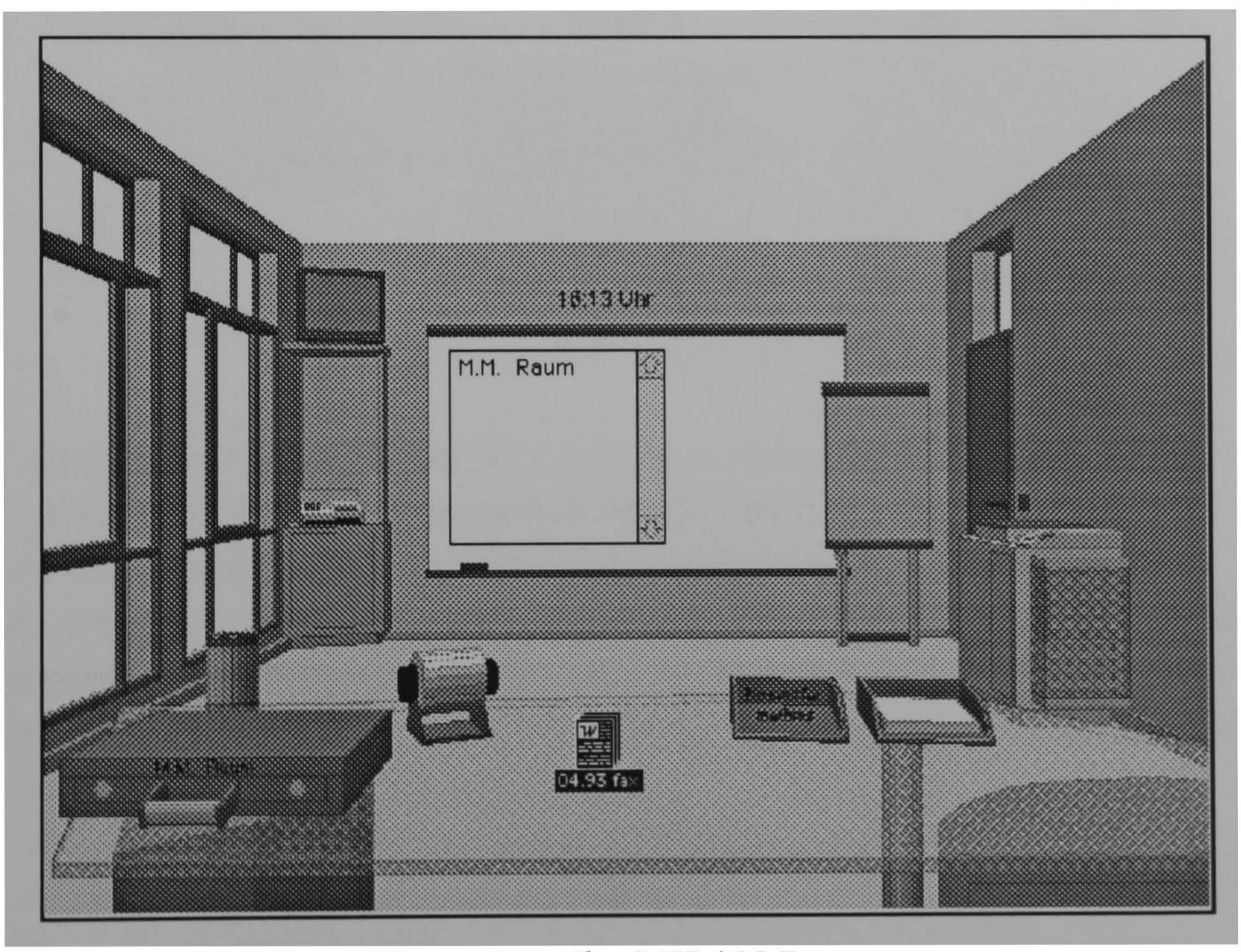


Figure 4.2: The MILAN Room.

A new version of MILAN was created to emphasise the spatial appearance of the interface. This was most notable in the redesign of the room, developed in a 3D CAD package, Virtus Walkthru (Virtus 1997), and presented to the user in perspective 2D. Three features were employed in this experiment, each represented by an object in the room: the out-tray for e-mail, the whiteboard for shared drawings and the television for the video connection.

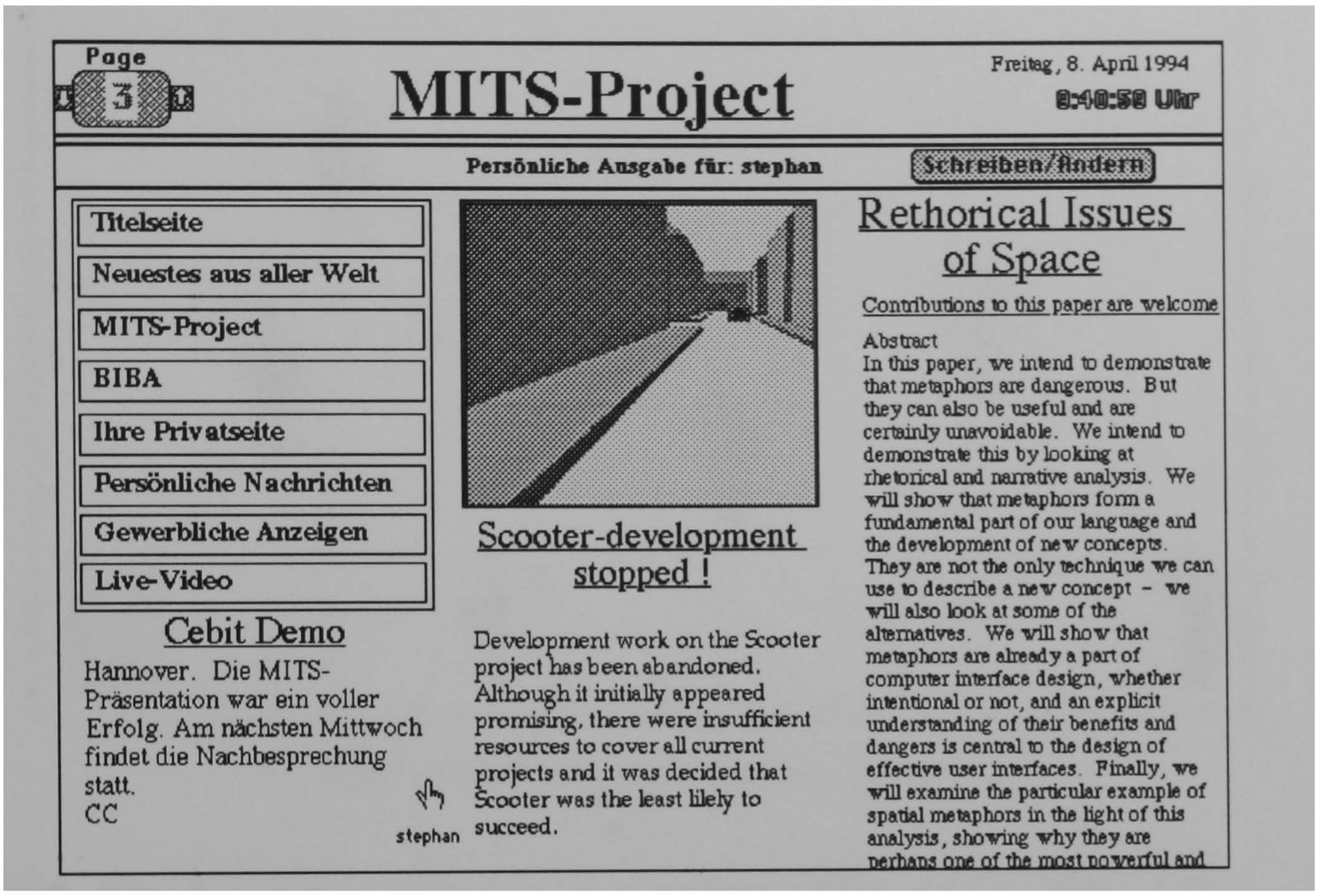


Figure 4.3: A group-page of the Link-Journal.

Looking like a DTP-Program, Link-Journal mixes the roles of editor and reader. It is divided into sections with different aims: a personal section accessible only by the local user; group sections which can be read and edited by members of a specific interest group; and public sections, usable by anyone who logs onto the system. The shared drawing was presented to the user as a group page of the company described in the subject's task. Leaving messages for other users (e-mail) was translated into a fill-in-form for personal ads in the paper. The video connection was implemented by dragging the picture of the required person into the 'live' picture on the page.

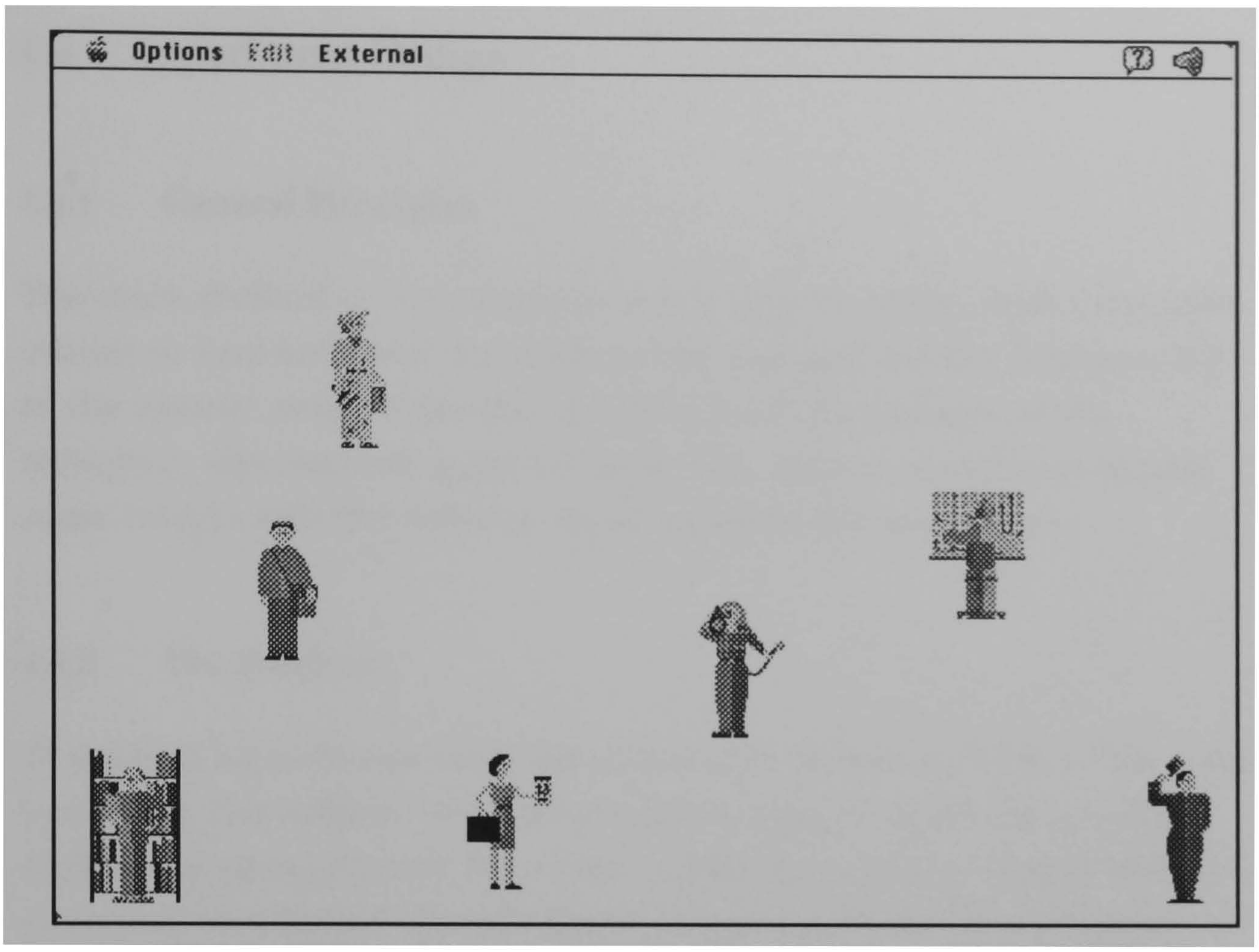


Figure 4.4: Little People Main Screen.

Little People displays different characters, or agents, on the screen each representing a specific set of actions. The three functions required for the experimental task were each represented by a different person. The postwoman sends e-mail, the cameraman controls the live-video and the designer gives access to the shared drawing tools.

The degree to which the interface itself could be manipulated spatially varied significantly between the three interfaces. In MILAN, all spatial relationships are fixed, as these provide the underlying rationale for this spatial interface, and nothing can be moved. In Link-Journal, the newspaper page provides a fixed space within which users can do no more than re-arrange the existing text and picture areas, or create new ones within the sort of constraints typical of a DTP program. In Little People, however, users had complete freedom to drag the icons around the screen.

4.6 Experiment Design

4.6.1 General Principles

The main method of investigation was a questionnaire, with questions chosen to find how well the subjects had mapped out the functionality of the system, even where this deviated from the domain of the metaphor. Open-ended questions were also used in an attempt to gain some insight into the subjects' thinking about the metaphors.

4.6.2 The Subjects

33 subjects were chosen from the staff and associates at BIBA, 11 for each metaphor. The subjects were of both sexes, aged 17 to 60 and covered a wide range of experience. They were chosen to represent typical factory personnel, but biased towards future requirements based on current trends. They included managers, apprentices, shop floor workers, secretaries, CAD operators and students of manufacturing design. The subjects had varied experience with computing, ranging from people who had never used a computer to experts, but were biased towards experienced users, as computer literacy is generally growing more widespread. Attempts were made to match users across the three interfaces but it was impossible to exert full experimental control over this.

The spread of ages and computing experience is shown in the diagram below. Age is shown in years, computing experience on a scale of 0 (the subject has no previous experience of using computers) to 100 (computer use forms a major part of the subject's daily activities):

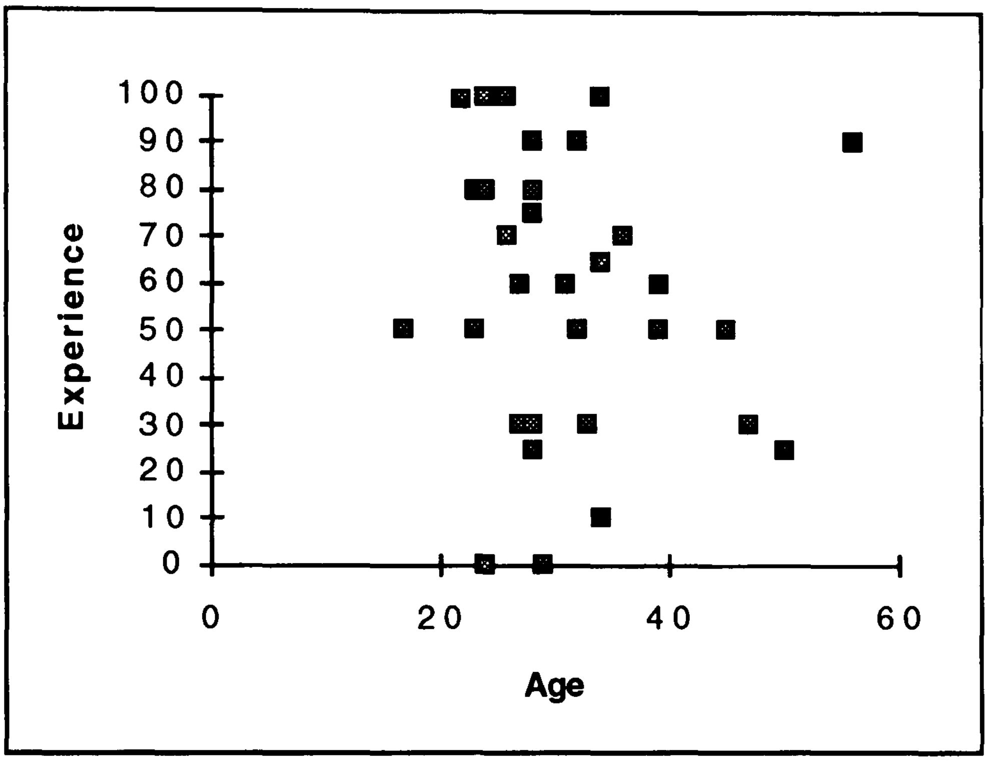


Figure 4.5: Age and computing experience of subjects.

Before starting the full experiment, a number of dummy runs were carried out with users who did not take place in the experiment itself. Subjects unfamiliar with the Macintosh interface had problems with some actions which were standard across all three interfaces. All subjects were therefore given written instructions on how to close windows, a short demonstration of dragging a mouse and a warning about waiting a few seconds for the system, which was sometimes rather slow, to respond to an action.

4.6.3 The Questionnaire

The questionnaire was presented to the subject immediately following successful completion of the scenario (no subjects failed to complete the tasks). Subjects were asked to sit in another part of the room for this so that they were unable to see the screen when answering the questions. Equal numbers of questions were chosen from the four categories based on the intersection of the user's model of the system domain and of the metaphor's domain. All questions were statements which the user was asked to judge as true or false. A 'correct' answer was one which showed that the user understood the functionality of the system, even where this deviated from the implied functionality of the metaphor. For example, the questions used for the MILAN system included the following, where **S** is the system and **V** the metaphor vehicle:

S∩V	Present in the system and implied by the metaphor vehicle: "You can see who else is in the room" (True).
S∩~V	Present in the system but not implied by the metaphor: "You can tell who is knocking on the door of a room you are in" (True)
~S∩V	Not implemented in the system but implied by the metaphor: "You can move the furniture" (False).
~(S∩V)	Not implied by the metaphor nor implemented in the system: "You can make a connection using a person's phone number" (False).

Statements were phrased so that half of them required the answer 'false', half 'true' for each category. The questions were then randomly mixed up, so that the categories would not be apparent to the users.

4.7 Results

4.7.1 Quantitative results

The amount of time it took the subjects to complete the scenario was less than expected from the previous experiment, where subjects took longer to complete a simpler scenario (Anderson 1994). Mean times taken for each interface (in minutes) were as follows:-

MILAN	08:40
Link-Journal	07:38
Little People	09:06

The variations in time between the three interfaces were not tested for significance, given the very wide variation in time taken by individuals within each category (standard deviation approx. 3 minutes for each category). There was little variation in the numbers of correct answers to the questions:-

MILAN	47% correct
Link-Journal	47% correct
Little People	57% correct

As random answers would have generated a score of 50%, it can be seen that, on average, the users did not form accurate mental models of the system functionality. No attempt was made to analyse these results any further. Even if the difference between Little People and the other interfaces is statistically significant, it is too marginal to be useful. This corresponds to the widely accepted distinction between *statistical* significance and *clinical* significance discussed in Sidman (1960) and Hersen and Barlow (1976). Robson (1993, p351-2; p367) discusses these views and those of Meehl 'who has claimed that reliance on statistical significance was one of the "worst things that ever happened in the history of psychology"' (Meehl 1978, quoted in Robson 1993, p351). It is not necessary to accept such an extreme view to see that the difference between the scores above is too small to provide useful guidance for an interface designer.

A more detailed analysis of the results is given in Appendix C. It should be noted that the results offer no support for the concept of conceptual baggage. This concept depends on accepting that users form mental models and that the accuracy of these models matters. It could be argued that this is because all three vehicles were conceptually richer than those used in the previous experiment. However, according to the concept of conceptual baggage, a conceptually rich vehicle should introduce more mismatches in the $\sim S \cap V$ case than in the others, an effect that was not found in this experiment.

4.7.2 Qualitative results

The questionnaire also included open-ended questions to ascertain how the users felt about the system. These were analysed to confirm the validity of the original metaphor classification. According to this, it would be expected that a user of MILAN would talk more about spatial relationships, a Link Journal user in terms of human interaction and a Little People user in terms of activities:

Spatial These were mainly comments about the positions of objects within the world represented by the interface (rather than simply the on-screen position).

Interactional Interactional aspects were separated from pure communication (see below). These were only comments on people *working* together or collaborating.

Activities All mentions of 'activity', 'function' or '*beruf*' (this approximately translates as trade or profession but bears a stronger implication of a specific activity).

In order to strictly distinguish these categories, some additional categories were included in the analysis:

Metaphor Mentions of the specific metaphor chosen. It was important to exclude comments in this category from those above, otherwise the exercise would be self-justifying. For example, identification of the relevant icon as the 'postwoman' was placed in this category and not counted as a reference to a trade or profession.

Technical Comments on sound quality, system responsiveness, etc. As a prototype system, response times were poor and sound quality was not always very good. However, none of this was relevant to the issues under consideration.

Communication Technical communications, rather than comments on co-working, which would fall into the interactional category. For example, comparison with videotelephony or mention of computer, video and audio working together.

Task The task in the scenario, i.e. changing the design of the chocolate box, rather than the general comments about activities as classed above.

Interface Mainly comments on user-friendliness, etc. This category also included mention of the physical layout of menus and graphics, which needed to separated from comments on spatial relationships.

To avoid bias and possible linguistic difficulties, the task was handed to a fellow researcher who spoke German as a first language and had not been involved in choosing the categories. Figures given are the total number of subjects making a statement falling into a category: some mentioned more than one aspect of it. No consideration was given to whether the mention was favourable towards the system or not, only whether the subject felt that aspect worth mentioning:

Table 4.2: Categories of responses.

System	Spatial	Interactional	Activities	Metaphor	Technical	Comms	Task	Interface
MILAN	**5**	1	0	10	0	4	0	9
Link-Journal	0	**7**	0	3	2	2	1	3
Little People	1	4	**4**	8	3	1	0	6

The three response categories that directly respond to the metaphor categories have been highlighted. Apart from the three main comment categories, it is noticeable that fewer people commented on the Link-Journal interface or metaphor, despite the fact that this was more dramatically different from their general working environment than the other metaphors. This is considered in the conclusions below.

In each of the three main response categories the corresponding metaphor type scores much more highly than the other metaphor types. These can be compared to the statistically expected frequencies:

Table 4.3: Expected frequencies.

	Spatial	Interact	Activity	Total
MILAN	1.64	3.27	1.09	6
Link J	1.91	3.82	1.27	7
Little P	2.45	4.91	1.64	9
Total	6	12	4	22

A χ^2 test is the commonest measure of statistical significance for this type of table but is only valid where all expected frequencies are above 5 in each cell (Siegel 1988, p.123). Siegel & Gallagher's recommendation for a smaller sample such as this is to use the Fischer exact test. However, the Fischer exact test can only be used on a 2X2 table. The experiment was therefore considered as three paired experiments. This was valid as each of the three experiments was carried out independently of the other two.

The Fischer exact test was applied to each pairing in turn:

Table 4.4: Fischer exact test results.

	Spatial	Activity-based	Total
MILAN	5	0	5
Little People	1	4	5
Total	6	4	10
p= 0.0238			
	Spatial	**Interactional**	**Total**
MILAN	5	1	6
Link-Journal	0	7	7
Total	5	8	13
p= 0.0047			
	Interactional	**Activity-based**	**Total**
Link-Journal	7	0	7
Little People	4	4	8
Total	11	4	15
p= 0.0513			

Accepting the convention that $p < 0.05$ is significant and $p < 0.01$ as highly significant:

the distinction between MILAN and Link Journal is highly significant;

that between MILAN and Little People is significant;

that between Link-Journal and Little People is not significant (though borderline).

4.8 Conclusions

4.8.1 The quantitative responses

This experiment aimed to test one assertion and ask a number of questions posed in Chapter 3. The first three of these are dealt with by looking at the quantitative results obtained from the main body of the questionnaire:

Question 1	Does it matter whether users form accurate models of the system?
Question 2	Would 'real world' users behave differently?
Question 3	Are the results valid for more complex computer systems and interfaces?

The second question has been very simply answered. None of the users was unable to complete the tasks in the scenario, despite some of them never having used a computer before. In their comments, subjects also claimed to find the system easy to use. Given that they were provided with no training, manuals or help facilities, the short time they took to perform the tasks demonstrates this quite convincingly. It was therefore concluded that the make-up of the user group had no effect on their successful use of the systems.

No breakdown of the times taken according to experience or other factor was attempted. Observation of users and the remarks they made during the experiment showed that their strategies varied considerably. Some expressed interest in the novelty of the interface and the facilities offered, exploring it thoroughly before starting to work through the scenario; others started the scenario tasks immediately. It is not clear whether this exploration time should be included in the time taken to complete the task. It is included in the times given above but accounts for much of the very high variance.

The first and third questions cannot be separated. It is obvious that forming coherent, overall, mental models of the system was not a condition for successfully using the interface, but the reason why these were not formed is less clear. It could be because of the greater functional richness of the system than that used in the study by Anderson et al (discussed in section 3.4) or the greater complexity of the interfaces. Certainly it is possible to say that inexperienced users working with a complex system were able to complete the tasks successfully without forming coherent mental models they could reason about. As corollary to this, there was no evidence of conceptual baggage.

If we believe that all users *always* generate mental models of the system, we have to conclude that these models were sufficient for efficiently completing the tasks, but the poor responses to the questionnaires demonstrate that subjects could not use their models to reason about the system as a whole. There is one possible explanation of this that remains consistent with the mental model view. In the case of MILAN, even though all users discovered that the television controlled the video and that the whiteboard was for shared editing, there was no need for them to integrate these separate objects into a coherent functional model of the total system.

When answering the questions, the subjects reasoned only about what they had to do at any moment to accomplish their tasks. The aim of the experiment was presented to the users in a task-oriented manner (making the changes to the chocolate box), so that the users built **action-oriented** mental models (Young 1983) which were not amenable to reasoning. This could lead to misinterpretation of some questions. For example, all but one of the subjects marked "true" for "It is possible to leave a message for someone without entering a room" (MILAN questionnaire) When carrying out this task, the users had been 'inside' a room. Even though they used the out-tray to send the message and should have recognised that out-trays were only present on the desks in the rooms, this was not relevant to the task at hand and did not feature within that part of their models of the system.

Thus, it is possible to maintain a view of human computer interaction based on the manipulation of mental models. However, to speak of the user forming a single, coherent model of the system is almost certainly wrong, but without such a model it is difficult to imagine how this approach could be used to develop a coherent model of the metaphor process. Conversely, the phenomenon of a user simultaneously holding a number of separate views of the system ties in well with the view of the system as a semiotic system leading to multiple signification.

4.8.2 The qualitative responses

Although the classification correlates significantly with the way that people perceived the systems, this does not mean that they identified the three classes in the same way. The responses of the MILAN users talked more of the metaphor and of the interface, and their spatial references were almost entirely about the layout of the room and the objects within it. The users had obviously formed a clear mental model of the interface, even if they failed to form one of the underlying system. For example, two users complained that the television was too far from the desk and one asked for a remote control (though this distance exists only within the perspective of the picture).

By contrast, the Little People users were more concerned with the functionality of the system, most notably the communication functions. Finally, the users of Link-Journal talked of interactional aspects in terms of the tasks that the system could support: cooperative working. In summary, the choice of metaphor did not only influence the user's view of the system; it far more fundamentally affects *what* the user sees, not just *how* it is seen: the level of signification.

The distinctions were particularly noticeable in the answers to the question asking what users thought the *Grundidee* (basic idea) of the system was. In the case of MILAN, almost everyone mentioned the metaphor. With Little People a more typical answer was 'presenting the functions of the system in a user-friendly way', whereas with Link-Journal, people frequently wrote of *Zusammenarbeit* (working together). In other words, MILAN users were most conscious of the *interface*, the immediate signification of the images on the screen (you are in a room, the television is on the other side of the room, etc.). Little People users were more concerned with the next level of signification, *what* the interface is *for*, i.e. supporting the *functionality* of the system (sending mail with the postwoman, setting video views with the cameraman, etc.). The Link-Journal users were concerned with a level higher still, *what* the functionality is *for*, i.e. supporting *people working together* (*Zusammenarbeit*, distributed manufacturing, etc.).

This confirms assertion 3 and answers question 4 positively:

Assertion 3	Interface metaphors create many different forms of signification not accounted for by the mental model approach.
Question 4	Does the type of metaphor affect the forms and levels of signification?

Although Link-Journal led to 'higher' levels of signification than the others, care should be taken before describing one approach to the system as 'better'. The best choice of metaphor will depend largely on what one wishes to get across to the users. Although the interactional interface (Link-Journal) appeared to turn the users' attention towards a 'more important' signification (what the system is to be used for), this is not always the first concern of the interface designer. For a system that is to be used for a short time, for example to support work groups that come together for short tasks before disbanding, it may be that the immediate appeal of an interface such as MILAN, in which the interface and the metaphor are foremost, is more important.

Although this moves beyond the general argument of this thesis, Appendix E builds on this experiment to examine the potential economic impact of metaphor choice, examining which type of metaphor is likely to be most successful in different industry sectors. This confirms that an interface based on an interactional metaphor is likely to be best in most industry sectors but that a spatial metaphor might be more useful in some. The number of interfaces used in the experiment (one interface based on each type of metaphor) is certainly not large enough to state this as a general case and it does not form a significant part of the conclusions of this thesis.

There is a common assumption that if user requirements and usability criteria are both met then users will use the services provided. There is considerable evidence that this is not always true. For example, Hutchinson & Rosenberg (1993) show that expert systems which meet identified needs and which are initially used by the users (implying reasonable usability) are then abandoned. Although they suggest other reasons, the results of this experiment suggest that it could be because the interface failed to 'sell' the system in the most appropriate way to that user group.

4.9 Summary of conceptual model

Chapter 3 proposed a semiotic model of HCI based on Layers of Signification (LoS). The studies described in this chapter have then checked the validity of the assumptions on which that model was based; the following chapter will then examine whether that model is effective and appropriate for use by designers. Before doing so, I will briefly summarise the main features of the model as checked by these preliminary studies.

Use of trope analysis has established that metaphor is ubiquitous in the computer interface, even where there is no apparent intention on the designer's part to employ it. This is consistent with the fact that, using the definition proposed in Chapter 3, the interface can be regarded as a semiotic system, consisting of related signs which affect the signification of other signs according to context. For example, the file saved by the 'Save' command in Word or WordPerfect depends on which file is in the active window.

Semiotics proposes that a sign consists of a signifier (the observer's immediate perception of the sign) which carries many significations, each leading to a separate signified. The simplest signification is known as denotation but even this is dependent on the observer – in English 'Gift' means 'present', whereas in German the word *'Gift'* means 'poison.' Higher levels of signification, also known as connotation, will be dependent on many other social and psychological factors. Analysis of the Macintosh interface established that it is possible to uncover examples of this multiple signification in a single interface element.

This multiple signification implies that the mental models proposed by some researchers will be inadequate to explain the user's interaction with the computer, in that they assume that the user should build a complete and consistent model of the system. A semiotic approach suggests that users will often be aware of contradictory significations within a single sign, making such consistency impossible. The use of metaphor then compounds this complexity by introducing all the layers of signification which the user associates with the metaphor vehicle.

If an interface element or the metaphor vehicle used in its construction leads the user into inappropriate signification, the user's understanding and acceptance of the interface can be severely compromised. This will be particularly important when the user and the designer come from very different social groups. Examples might include educational level, profession, sex or age, all of which will influence higher levels of signification. Section 3.6.6 proposed interviewing users of computer interfaces, continually asking, "What is that for?" in response to their answers. It is proposed that this 'What for?' interview might help the designer to uncover some of this signification. The following two chapters will pursue this further.

5 Methods

5.1 Selection of methods

5.1.1 Introduction

As stated earlier, the aim of this work is to explore the role of metaphor in human-computer interaction and, hopefully, to provide assistance to interface designers in their choice or use of metaphors. According to de Saussure and Barthes' theories, the signification of interface metaphors to the user will exist on many levels. Other research and my own preliminary studies also suggest that interface metaphors play many different roles in the interaction of the user and the system. Metaphor is used as a means of introducing novel concepts to the user but also brings with it conceptual baggage. It can turn users' attention towards different aspects of the system or towards their own purpose in using the system. The metaphor vehicle also introduces concepts which themselves carry many levels of signification, above and beyond the metaphor's immediate support for user.

Each of these aspects of interface metaphor might provide a valuable area for further investigation. This provides a wide range of potential methods taken from many different fields of study, including linguistics, semiotics, psychology and sociology.

5.1.2 Potential methods

Some of the possible research issues and potential methods for their investigation are summarised in the following table. The table is by no means exhaustive but lists some of the principle research methods which might be worth consideration:

Table 5.1: Potential research methods.

Research issue	Field of study	Research methods
Metaphor as human-computer interaction.	HCI, software ergonomics.	Experimental psychology.
The role of metaphor in the user's mental model of the system.	Cognitive psychology.	Experimental psychology, computer modelling.
The role of metaphor in the user's motivation and work effectiveness.	Management science.	Surveys, interviews, case studies, economic analysis.
The cultural role of interface metaphors.	Anthropology.	Ethnology, case studies.
Interface metaphors as a social artefact.	Sociology, social psychology.	Observation, interviews, action research.
The mechanism of metaphor.	Linguistics, rhetoric.	Rhetorical analysis, grammatical analysis.
The interface metaphor as a sign.	Semiotics.	Semiotic analysis or de-construction.

Chapters two and three drew on literature from a number of fields which offer possible approaches for this study. As the quotation from Whittock (1990) in Chapter 3.4.1 pointed out, the most obvious approaches to the study of rhetoric are either those based on cognitive psychology or those of 'rhetoric and strategies of communication'. I have carried out limited investigations in both these areas, as described in the previous chapter. The experiment based on cognitive models indicates that metaphors can operate at different layers of signification leading to very different relationships of users to the system and their purpose in using it. However, it would be difficult to extend this experiment to a more rigorous comparison of metaphor classes without a very large number of implementations of many different metaphors which would be well beyond the scope of this thesis. However, this does not rule out other methods of examining the phenomenon of multiple signification more deeply.

Rhetorical analysis is normally used to examine the content of a text and categorise the various tropes and schemes used. Although this was useful in demonstrating the prevalence of metaphor and metonymy in the MS-DOS command language, it does not offer any assistance to the designer nor provide any insight into the human aspects of human-computer interaction. Another method based on a 'strategy of communication' is semiotic analysis, although this is far more subjective, de-constructing a text to draw out its full signification to the reader. It could be questioned whether it is appropriate to use such a subjective method as part of a thesis in the field of HCI.

Burrell and Morgan (1979, p. 6-7) grade research methodologies in a continuum ranging from the nomothetic to the ideographic. Nomothetic methodologies are deductive and objective, characterised by systematic protocol and technique; ideographic methodologies are inductive and subjective, characterised by 'getting inside' situations. According to this categorisation, semiotic analysis is overwhelmingly ideographic. The associated subjectivity is not a fault of semiotics but a central feature, in that the semiotic viewpoint sees all meaning or signification as subjective, as is made clear by the semiotic view of the relationship between the signifier and the signified which takes place entirely within the head of the reader or observer, who 'reads meaning into the text'. The analyst thus has to attempt to 'get inside' the head of a potential user.

Robson (1993, p.18-19) makes a similar distinction between 'scientific' and 'interpretive' approaches. He points out that the former is often described as 'hypothesis testing' and the latter as 'hypothesis generating.' However, he goes on to say that, "many of the differences between the two traditions are in the minds of philosophers and theorists, rather than in the practices of researchers." (Robson 1993, p.20). He quotes Bryman in support of this view:

> The suggestion that quantitative research is associated with the testing of theories, whilst qualitative research is associated with the generation of theories, can... be viewed as a convention that has little to do with either the practices of many researchers within the two traditions or the potential of the methods of data collection themselves. (Bryman 1988, p.172)

In the preface to his book, Robson (1993) admits that, as an experimental psychologist, he "started with a virtually unquestioned assumption that rigorous and worthwhile enquiry entailed a laboratory, and the statistical analysis of quantitative data obtained from carefully controlled experiments." However, his interest in real world research demanded approaches which could "say something sensible about such complex, messy, poorly controlled 'field' settings." In his case suitable, though more subjective, methods came from the sociologists and social psychologists he worked with. Semiotic analysis of a user interface, however, does not describe how that interface is viewed by its users in the real world, rather it looks at all the possible ways in which it might be viewed by users. Semiotic analysis could be used to deconstruct the language of users' interaction with their systems but this would be a much more extensive task than analysing the interface and it is questionable whether it would yield as useful results as ethnographic approaches which have been developed explicitly to study such real world interaction.

A fourth approach was proposed in Chapter 3 – that of the 'What for?' interview technique. Such a use of simple open-ended questions is known as 'probing'. It offers a simple technique which designers could employ with their own users and, though it is related to the semiotic method, it is more formalised, leaving less room for the designer's personal bias and obtaining data purely from the user.

5.1.3 Probing

Probing was developed as a technique for use in a particular form of non-directive interview – the focused interview (Robson 1993, p.240-41; Zeisel 1984, p.140). Rubin and Rubin identify three reasons for using probes:

> Probes encourage the speaker to keep elaborating. Second, probes ask the interviewee to finish up the particular answer currently being given... The third function of probes is to indicate that the interviewer is paying attention. (Rubin 1995, p.148).

They then identify five types of 'housekeeping' probes: elaboration, continuation, clarification, attention and completion.

> They ensure that you are getting a reasonably accurate and understandable answer while encouraging the interviewee to keep talking. But probing does more than keep the conversation going, it helps get the depth and dependability you need. (Rubin 1995, p.150).

Rubin and Rubin also describe steering (p. 208), sequence, experience, evidence and slant probes (p.208-10). These are not relevant to this experiment because, as their names indicate, they are used by the interviewer to steer the interview in a particular direction whereas 'What for?' probes are intentionally non-directive.

Zeisel (1984, p.141-56) provides a more detailed analysis of the types of probe an interviewer might use, categorising them as follows:-

Addition probes to promote flow - used to get respondents to express themselves more fully, e.g. encouragement and body language - "I see" or a nod of the head.

Reflecting probes to achieve non-direction - echoing the respondent or responding to a question by repeating it back.

Transition probes to extend range - moving on to the next issue or expanding an issue that was mentioned but skipped - "that reminds of something you were saying earlier" or "that raises the general issue of..."

Situation probes to encourage specificity, e.g. pointing to a map or a picture to establish precisely what the respondent is talking about.

Emotion probes to increase depth, e.g. "what do feel about this?"

Personal probes to tie in context, e.g. "is there anything particular about you that makes you feel strongly about this subject?" "does that relate to some previous experience you've had?"

The closest of these categories to the type of question used in the 'What for?' interviews is that of reflecting probes. Certainly, the 'What for?' probe supports non-direction. However, in interviews quoted at length, Zeisel includes another category which is even closer - the general probe. For example:

Respondent: I am afraid to live in that area

Interviewer: What are you afraid of? (p.153)

Respondent: I find this office extremely inefficient and wasteful.

Interviewer: In what way? (p.155)

These examples represent the closest category of probe to the type of question I am advocating. Zeisel's book is specifically about the use of interviews to gather respondents' opinions about environmental situations - where they live, work or visit - in order to inform design and planning decisions. This is comparable to using such techniques to get information about an interface the person works with to assist in the design of that interface. The difference between the 'What for?' technique and Zeisel's focused-interviews lies in the role of the interviewer. In his case there is a specific focus towards the design issues, whereas the 'What for?' technique simply attempts to uncover as many layers of meaning as possible; it is for the designer to consider whether these are relevant afterwards.

5.1.4 Research validity

Gill and Johnson (1997, p.128-29) offer a number of criteria by which the validity of a chosen research method might be assessed:

> *Internal validity.* The degree to which the researcher can be sure that the 'cause' is what actually produces the effect.
>
> *External validity.* The extent to which the research can be generalised. This can be subdivided into the following:
>
> > *Population validity.* The validity of generalising from the research sample to the population in general.
> >
> > *Ecological validity.* The validity of generalising from the social context of the research to other contexts and settings.
>
> *Reliability.* The consistency of the results and the degree to which another researcher would be able to replicate the original research.

As consideration of cause and effect is not relevant to exploratory research (see Section 5.2.1), internal validity will not be considered at this point. However, it is also necessary to consider the practicality of the methods and whether they could yield 'useful' results to help interface designers, giving the following table:

Table 5.2: Validity of research methods.

Method	Populat'n validity	Ecological validity	Reliability	Practical-ity	Useful-ness
Rhetorical analysis	N/A	Low	Medium	High	Low
Semiotic analysis	Low	Low	Very low	High	Medium
Experimental psychology*	(Medium)	Low	(Medium)	(Low)	Medium
Probing	Medium	Medium	Medium	High	High

(*In the case of experimental psychology, there is an inverse relationship between the validity criteria shown in brackets and the practicality of the experiment. As explained above, comparing metaphor categories with moderate population validity and reliability would require a great many experiments, giving a low level of practicality.)

Rhetorical analysis can be excluded as not giving very useful results, while semiotic analysis could give useful results but must be rejected as its external validity is so poor. By contrast, the potential of probing with the 'What for?' technique could be demonstrated by a relatively small number of interviews, with the potential to yield highly useful results of reasonable validity. This approach therefore formed the basis for the research design described below.

5.2 Research design

5.2.1 The purpose of the enquiry

The purpose of this enquiry is not to uncover useful information about specific metaphors or categories of metaphor, but to find out whether 'What for?' interviews offer a potentially useful technique for interface designers to use. Robson (1993, p.42) distinguishes between three principal purposes of enquiry: exploratory, descriptive and explanatory. Of these, investigation of the 'What for?' technique falls into the exploratory category which he characterises as follows:

- To find out what is happening.
- To seek new insights.
- To ask questions.
- To assess phenomena in a new light.
- Usually, but not necessarily, qualitative. (Robson 1993, p.42).

Robson points out that it is commonly suggested that there is a hierarchical relationship between the research strategy and the purpose of enquiry:

- Case studies for exploratory work.
- Surveys for descriptive studies.
- Experiments for explanatory studies. (Robson 1993, p.43).

While accepting this as a general rule, Robson points out that it is not absolute – for example, case studies have been used for all three purposes. In considering the 'What for?' technique, some form of case study does indeed appear to be most appropriate. The technique is intended to uncover the higher levels of signification which depend on the context – *what* the user is using the interface *for*. This context would change radically in the laboratory where a user would be using the interface to help in an experiment and the signification would be radically different. To test the technique it is therefore necessary to use it in the real world, as close to the conditions in which a designer might use it as possible.

Conventionally, both case studies and surveys examine what is happening in the real world. This is obviously not possible in this case, as the technique is not yet being used. The research must therefore take the form of one or more case studies in which the technique is taken into the real world and applied within it. As such it forms what Robson (1993, p.41) classes as a hybrid strategy, combining aspects of quasi-experiments and case studies.

5.2.2 Interview structure

Uncovering the signification of the interface to the user means any interview must be user-directed – the interviewer must not ask any leading questions. The nature of recursive signification introduced in Chapter 3 implies that the interviewing technique should also be recursive.

The unstructured semiotic analysis of part of the Macintosh user interface, discussed in Chapter 4, section 2, looked at how a user might 'read' an interface. The model above represents a clearer way of encouraging the user to articulate his or her signification:

- Keep asking "Why?" or
- Keep asking "What for?"

Technically, there is a difference between these two questions in some circumstances. 'What?' implies an object, action or concept. In the context of the sign then, if a *signifier* exists (which it must do to ask the question), the object will be the *signifier*. 'Why?' implies a mechanism, in this case the *signification*. In practice, the main constraint is the nature of the English language which favours one construction in some cases but not others. For example, it is more meaningful to ask, "What is a spade for?" rather than, "Why is a spade?"; "Why is the sky blue?" rather than, "What is the sky blue for?"

However, this is not an absolute rule in our everyday use of English. For example, we would usually ask, "Why did the chicken cross the road?" rather than, "What did the chicken cross the road for?" In many circumstances, the two questions are interchangeable: "Why did you do that?" is directly equivalent to "What did you do that for?"

These principles form the basis of the interview technique. The first preference is to ask, "What for?" rather than "Why?" There are two reasons for this. A simple one is that the interview must begin with "What for?" because the interviewer does not know the user's initial *signified* at that point. For example, the interviewer might ask, "What is that for?" but could not ask "Why is that?" At a later stage the interviewer could ask "What do you use that for?" rather than "Why do you use that?" but only once the interviewee has made it clear that he or she does use the interface element referred to.

The second reason for preferring "What for?" is that the user is more likely to be aware of the signified than the signification and to answer questions in those terms. Sometimes the 'What for?' question is difficult to phrase and 'Why?' is easier and carries the same meaning in normal conversation. Asked what an interface element is used for, the user might answer, "to send reports to headquarters." The question "Why is that used?" would receive the same reply. Technically it could be answered, "because it is labelled 'reports'" as this is its immediate signification, but this type of response did not occur in the pilot studies. Whichever form of question was asked, users would answer with *what* they send reports to headquarters *for*.

One exception to this is where a interviewee replies that there are many answers to the question. The interviewer can then pause to see whether the interviewee follows up with examples or probe for them with the simple question 'such as?'. Although this may seem to limit the user, it is not proposed that the 'What for?' technique provides information on all possible lines of signification that a user might take. If the interviewee considers that the other significations are important, he or she can return to them later in the questioning.

5.2.3 Pilot study

The main aim of the pilot was to check that the technique would be likely to work and to gain skill in interviewing. Both users and designers were interviewed as it seemed that the technique might raise some interesting contrasts between the signification for the two groups. All of the designers and some of the users were friends or relatives and thus well-known to me and unsuitable as subjects for the main experiment, but adequate to check out the technique and decide whether this direction was worth pursuing. Systems were studied across a range of applications and user environments to see whether this appeared to affect the applicability of the technique.

Interfaces 1 and 2 were both developed within the IT support team for a Local Education Authority (LEA). One interface considered was a statistical reporting system developed in Excel and running on a PC. It is used to account for the placement of special teachers to support children who do not have English as a first language and to report back to the Home Office. The second system runs on an IBM AS/400 and supports a form-based interface used to administer the payment of student grants.

Two interviews were carried out with designers of a manufacturing system. Unfortunately, the company was taken into receivership shortly after the interviews and it was not possible to gain access to users. The system provides feedback on scheduling for advanced manufacturing.

The fifth interface was a Web page set up by a fellow researcher in Brunel whom I interviewed as the developer of the interface. I also interviewed a research manager from a different research centre who had used the web site's diary facility to set up a meeting. At the time, neither researcher was aware of the details of this thesis.

For the pilot study, I carried out the content analysis myself. The bias inherent in this, together with the small number of interviewees, means that the results are not suitable for extensive analysis. The numbers of separate layers of signification uncovered in each interview are shown in the table below:

Table 5.3: Levels of signification in pilot study.

	Number of layers of signification	
Interface (sector)	Designer	User
1 (education)	12	13
2 (education)	9	9
3 (manufacturing)	10	N/A
4 (manufacturing)	7	N/A
5 (research)	12	12

As the table shows, similar numbers of layers were uncovered in every interview, across both designers and users and across usage sectors. This may well have been because, in all cases the users were personally known to the designers and, in all but the web interface, the interfaces formed part of bespoke systems designed for those specific users.

A number of interesting features were observed when examining the transcripts. Some interviewees started looping, going back to a previous answer and repeating the explanations given. Where this loop was obviously going to be repeated, I finished the interview. In one case, however, the interviewee backtracked and provided a new set of significations. Successive layers of signification led him to saying that he wanted a good job. When asked what for, he said it was for the money but then backtracked and gave the explanation that he was actually looking for personal fulfilment in his work.

Apart from the branches and loops, most responses started at what appeared to be the simplest levels of signification, such as 'it produces a report', progressing upward to higher motives such 'education is a good thing'. The only exception was the researcher who had used the Web page. Possibly because he was used to looking at why people use systems and how they are structured, he began his responses by saying, "it's a link to another page in Netscape". He then attempted to give an explanation of people's underlying motives for using the Web in general before saying, "That's probably reached the end." He then added, "I've taken your questions in a general sense instead of looking at that particular page but then after all you did point me to that word 'diary'." He then began at the 'bottom' level, explaining why he had used the diary facility, until he reached the level of signification at which he had originally started.

The following quotations from the interviews show the highest levels of signification reached by each of the interviewees. Judgement of which level was the 'highest' was a purely subjective choice on my part:

Table 5.4: Highest levels of signification in pilot study.

Interface	Designer's signification	User's signification
Education sector		
1	Because it is a good idea to educate kids. There are political reasons.	Various political issues, concerned with under-achievement of the children
2	It is a good thing that people go to college to study.	[The government] have to encourage people to stay in education.
Manufacturing sector		
3	Quality of life in terms of earning salary.	N/A
4	It is a bad idea to have increased costs or late orders.	N/A
Research sector		
5	Using my mind and making the best use of my ability. To make me happy.	Exploration or interest in the back of my head

There are close similarities between the responses given by the designer and the user of each interface and between people working within the same sector. Again, this could well be because the people concerned work with one another and share a common viewpoint. It should be noted that each interviewee might also see other high level significations which would have been revealed in other interviews. However, it is noteworthy that most interviewees were able to relate the interview to concerns which are well beyond the normal considerations of interface designers, such as politics, morality and personal happiness. Only one interviewee raised a concern that might normally be considered by the designer: to reduce costs.

In summary, the pilot study was similarly effective for all the industry sectors and interfaces considered. Apart from the web page, there appeared to be little difference in the responses given by the designers from those given by the users. It should be noted that all these systems were designed for a small number of users and it should be expected that the designers would be familiar with the users' concerns.

5.3 Implementation

5.3.1 Choice of subjects and interfaces

Although the pilot studies included both designers and users, no useful distinction was found between the two groups. It would also be very difficult to gain access to the designers of interfaces for generic applications, as these are rarely designed by a single individual. As the 'What for?' technique is intended to help the designer gain useful information about the user, the full experiment was limited to this scenario and no interviews were carried out with designers.

No analysis tool can guarantee to yield useful information for all possible analysts in all possible interface design conditions. To establish the potential of the 'What for?' technique, a single interview might be enough to show that the technique *could* yield useful results. In practice, designers are only likely to use a technique where they consider that the information obtained is worth the time expended. A more useful test would therefore need to establish a 'reasonable case', such as interviewing users from at least two different user groups using different types of interface.

As the experiment involves the assessment of an interface element, this must either be part of an existing interface being assessed, an existing interface due for re-design, or a potential interface being assessed in prototype form. As it is more difficult to gain access to prototypes, an existing interface was chosen for both sets of interviews.

In considering the number of interviews, it is also necessary to consider the conditions in which a designer might use the 'What for?' technique. It is not suggested that the interviews would provide all the information a designer needs but that they should be one of the tools available for user requirements gathering. In practice the designer of an in-house system is constrained in user requirements gathering by the number of people who will use the system. This also formed a constraint on the number of users interviewed in my research. In the pilot study, most of the bespoke systems were used by four or five users, some by only one user and one by 'about twenty'. Whilst generic applications might be used by a much larger number of users, initial studies by the designer are likely to be limited to a similar scale. The results from the pilot study also indicated that this scale of study could yield interesting results.

The pilot study indicated that one factor likely to affect the signification of the interface to the user was whether the interface was part of a bespoke system or part of a generic application. It was more difficult to obtain access to users of bespoke systems but personal contacts were used to gain access to a group of users within a major communications company using an international accounting system. Although a friend provided my introduction, neither the designers of the system nor the users were previously known to me. The second interface chosen was that of Microsoft Word, one of the most widely used generic applications. For ease of access, the second user group was composed of doctoral students taken from the Department of Information Systems and Computing at Brunel University. The researchers were working in a number of areas of computing, principally information modelling. Given that the experiment lies in the field of HCI, researchers in this field were excluded from the user group.

One element was chosen from each interface to form the basis of the investigation. In each case, a frequently used metaphor-based interface element was chosen, although it is likely that the frequent usage had led to the death of the metaphor for both groups. In the case of the accounting system, the chosen element was the 'Navigate' command on the tool bar at the top of all screens, used for changing to a new screen; in the case of Word, it was the 'Save as...' command on the pull-down 'File' menu.

The first interface examined formed part of an international accounting system. The total number of users at the main site was five, all working at the same group of desks in the same room. As the other users of the system were at remote locations, mainly in Australia and the far East, it was not possible to obtain a larger sample than these five. The number of doctoral students interviewed was therefore also set at five to provide a balanced comparison.

5.3.2 Locations and times

The interviews with the users of the accounting system were arranged with the manager of the group to suit their availability - a factor over which I had no control. In the event, they took place in their normal workplace between two o'clock and four o'clock on a Monday afternoon. Their workplace is an open plan office which they share with five other teams, each of five to six people working in related business areas. To avoid disruption to the other workers or the chances of other interviewees over-hearing, the interviews took place in a small meeting room opening off the main office.

Interviews with the doctoral students using Microsoft Word were therefore arranged for the same time on the following Monday afternoon. Nine students were working in the same room, one of whom was known to me and therefore excluded. Of the others, five were immediately available for interview, and formed the user group for the study. The room is located in an attic area and it was possible to interview the researchers in a corner of the room without the other researchers being able to hear or see the activities.

Traditionally, methods such as the NCC (National Computing Centre) systems analysis and design methods stressed the importance of the analyst interviewing users in the users' workplace (NCC 1978, p.106-109). Newer methods claim to be heavily concerned with user understanding but this is expressed in terms of giving training or information to the user (Norman 1986b p.153-238) or of bringing users into the design team (Yeates 1991, p.18-28). This contrasts with the more traditional attitude in which the analyst would gather information from the users, going into the user's workplace to do so.

Whether this is deliberate or the importance of workplace interviews is simply taken for granted, the justification given by the NCC (1978, p.107) appears to remain valid – that interviewing in the workplace "can be an advantage, since the interviewee will feel more at home and additional information can be obtained from observation. Interruptions may tell a lot more than the interview itself." I have assumed that designers should continue to hold such interviews, and that the 'What for?' technique would form a part of them. I therefore carried out all interviews at the users' normal workplace. However, in both groups the interviewees worked very closely together, making it necessary to take each individual subject to a spare desk in one corner of the room or a side room during the interview itself to avoid others over-hearing the responses.

5.3.3 Interview practice

It was important that interviewees answered the questions freely without worrying about their remarks being taken as specifications of the software or complaints about it. It was also important that the interviewees were ignorant of the reasons for the questions (apart from their assistance in my PhD work), in order to avoid attempts at 'correct' answers. Finally, the pilot studies had shown that interviewees were sometimes bothered when they were unable to answer probes towards the end of the interview. I therefore read out the following paragraphs at the start of each interview (adapted slightly for the Microsoft Word users as only their personal anonymity needed to be assured):

> I would be grateful for your help in some research I am carrying out for my PhD. I will ask some questions which I would like you to answer as simply and honestly as you can, where possible with a single sentence. Your answers will not be treated as a specification of the software and will only be used for research purposes. Your identity, the identity of the software and of this organisation will remain confidential. When the interview is complete, I will send you my record of the interview which you may correct if you wish to do so.

> The questioning technique may seem a little unusual but I will be glad to explain its purpose once the interview is over. The technique is progressive and will probably lead to questions which you feel unable to answer. This is OK: please just say so and I will wrap up the interview.

Once this statement had been read and accepted, the next step of the experimental procedure was to point out the interface element forming the focus of the interview and ask 'what is this for?' The interviewee's response was then asked about in the same manner until the answers formed a closed loop or the interviewee felt that the question was unanswerable. In some cases it was necessary to repeat a question in a slightly different form when the user failed to answer. After the interviews were completed, a transcript was given to each subject to be checked for accuracy.

There was a risk of potential alienation of the interviewees which might reduce their cooperation if they were asked personal questions. Characteristics such as sex and age were therefore assessed by myself to avoid any chance of this happening. In the case of age, this consisted of placing people into the age groups: under 25, 25-35, 35-45, 45-55, over 55. These assessments, together with the other main characteristics of the two sets of interviewees are summarised in the table below:

Table 5.5: Characteristics of user groups.

	Group 1	**Group 2**
Occupation	Clerical	PhD students
Sex	Four female, one male	All male
Ages	25-55	25-35
Organisation	Large communications company	University
Location	Open plan office	Open plan office
Interface	Oracle-based accounting system	MS Word (wordprocessor)
Interface element	'Navigate' command on tool bar	'Save as...' pull-down menu command

5.3.4 Choice of personnel

As an experienced interface designer with training and experience in user requirements gathering, it was valid for me to carry out the interviews myself. Although I was biased in hoping the technique would uncover as much useful information as possible, any designer using the technique in the real world would also wish to uncover as much useful information as possible and thus have a similar bias.

In practice, it is probable that the analysis of the interview content would also be carried out by the interviewer but, for this experiment, bias might be introduced if I carried out the content analysis myself. The analysis of the interview data was therefore carried out by independent evaluators. To check on consistency, two evaluators were chosen, one a fellow researcher in HCI, with experience of user interface design, the other a media studies graduate with previous experience of content analysis. Neither was informed of the aim of the experiment beyond what was necessary to train them in the content analysis method.

5.4 Analysis

The interviewees' responses were broken down into elements (see below). These elements were then given to the two independent evaluators for content analysis.

5.4.1 Content analysis methods

Budd et al (1967) and Krippendorf (1980) describe a number of approaches to content analysis, ranging from quantitative analysis of large quantities of data through to qualitative analysis of small amounts of data.

At one extreme is the tightly controlled quantitative approach, such as frequency comparisons of specific words or phrases. This generally requires very large quantities of data for analysis to give statistically significant results.

A second, more common approach is a looser version of this, relying on subjective gathering of words and phrases into categories. For example, the first method might count references to 'press freedom', and perhaps 'freedom of the press', whereas the second approach might also include references to 'journalists' rights', 'censorship' and 'protection of privacy.' Neither technique will normally consider whether the references are favourable or not. For example, a factor might be how much interest the press of a country shows in a particular issue, but not its opinion on that issue.

The third form of content analysis is context sensitive. This relies on a further subjective assessment of whether a subject is mentioned in a favourable, unfavourable or neutral context. This is usually applied when an analysis is made of changes of attitude over time rather than providing an assessment of the actual balance of opinion at a point in time. It might, for example, show increasing support in the press for censorship or increasing support for press freedom. Commonly, it is based less on the statement of explicit opinions than on the form of language used, such as 'gagging order' (derogatory) versus 'privacy protection ruling' (complimentary).

Another form of content analysis specifically covers the analysis of ethnographic studies. I am not carrying out this type of study and did not consider this further. The final category is the one which is apparently the most appropriate for my work: content analysis of interviews or case studies. Unfortunately, this is covered least in the standard books and papers on content analysis.

The principal use of this type of content analysis is in the analysis of psychiatric case studies or similar types of interview. In this context, it is briefly touched on by Chirban (1996) and Gorden (1987). Neither of these authors gives any details on how to use the method which, it appears, is usually a matter of ad hoc design by the experimenter. I therefore returned to the standard content analysis texts of Budd et al (1967) and Krippendorf (1980), adapting their methods to fit the conditions of this study.

Analysis of the 'What for?' interviews is not content analysis in the conventional sense, in that there is no intention to seek pre-defined categories of signification. Content analysis usually depends on proving a particular theory through categories: "No content analysis is better than its categories, for a system or set of categories is, in essence, a conceptual scheme." (Budd 1967, p.39). However, the conceptual scheme in this case is much simpler - that a personal set of categories exists for the individual user and that these are hierarchical. The exact content of a category is not relevant to this, though it could form an interesting area for further research. The hierarchical nature of the categories is more difficult to prove, although it could be argued that it flows automatically from the recursive nature of the interview technique.

Krippendorf (1980, p.75-81) also places emphasis on the definition of categories, stipulating that categories must be defined by both definitions and examples if presented to an untrained observer. If applied to this work it would require the construction of 'extensional lists' (Krippendorf 1980, p.76-77) by which every expression within the text is given a tag to indicate its category. This is the approach taken, with numbered tags added to the response elements by the evaluators as the first stage of the content analysis process.

5.4.2 Structuring the responses

The chosen approach obviously depends on the splitting of the responses into elements to be tagged. Each response was split into sentences and again into clauses. These were further broken down into sub-clauses where a preposition or subjunction had been used which could potentially be used to introduce a new meaning, such as 'to', 'for' or 'and'. In addition, where there was any possibility at all that a separate meaning might have been introduced, the clause was split.

The training for the evaluators, based on responses in the pilot study, included both examples in which elements had to be further split into multiple signification and examples where consecutive elements had to be gathered together into a single signification. However, the breakdown of the responses in the main experiment was deliberately biased towards excessive splitting of the responses. A new element signals the possibility of a new signification but the evaluator can always group elements together, whereas spotting multiple signification within a single element will depend on the evaluator detecting the change in meaning.

Consideration was given as to whether the elements should be presented to the evaluators in random order. However, the context of the elements was necessary to disambiguate them. Consider, for example, the following responses:

Table 5.6: An example of different categories allocated to the same phrase.

Response element	Tag
...to bill one part of COMPANY...	8
...to another part of COMPANY.	8
Because if one part of COMPANY is doing work...	9
...or providing services...	9
...to another part of COMPANY...	10

In this example, the phrase 'to another part of COMPANY' was used twice but in different contexts: once to refer to billing and once to refer to the provision of services. It is necessary to present the elements in context to make this distinction clear. In context, the evaluator spotted that the second use of the phrase referred to a separate signification and tagged it with a different number.

Names and other details of the interviewees were removed from the response sheets, although names of the software and the organisations were left unchanged at this stage. Only the responses were included, not the 'What for?' and 'Why?' questions which had prompted them. The response elements were printed out in tabular form for each evaluator, with two additional columns, one for the tags and one for any comments.

5.4.3 The content analysis process

Each evaluator was given a brief training in the content analysis process. The concept of signification was briefly summarised and the evaluators were asked to indicate "wherever a new meaning was introduced." This was done by tagging each response element with a number, starting at '1'. Where two elements carried the same meaning the evaluators were instructed to give them the same number. Where an element contained no signification, such as 'I don't know' or 'the second reason is..', it was to be tagged with a '0'. I worked through one example from the pilot study and each evaluator then practised with another pilot study example under my supervision.

The actual content analysis then consisted of two phases. The first phase was for the evaluators to tag the response elements. Each of them went separately through each set of responses, tagging them with numbers as in the training example. The evaluators were invited to use the comments column to raise any questions or uncertainties but none were entered at this stage.

The second stage of the process consisted of using the tagged comments to indicate common signification between interviewees. The first set of responses was used to indicate the initial set of categories to be used for the analysis. All '0' tagged elements were removed and the remaining answers sorted into numerical order. Although this lost some of the original contextual information, most of it remained. Where elements had been consecutively numbered, they remained in the order of the original responses, as they did when a group of elements were given the same numbers. The contextual ordering was disrupted when interviewees had returned to a previous meaning but, in these cases, ambiguity was reduced by the multiple entries for that tag. In practice, neither evaluator expressed any difficulty in identifying the categories.

The second set of responses was treated in the same manner and the evaluator was then asked to compare it with the first set of responses and mark any duplicated signification across the two interviews. The results of this were then used to combine the two sets of responses into a single sheet of categories. The numbering of the first response set was maintained. Where no duplication of signification was marked, the second set of responses were tagged '1.1', '2.1' and so forth. This allowed the two sets of responses to be merged in numerical order, maintaining the original flow of the responses. The third response set was then marked by the evaluator where it duplicated any meanings contained in the combined set. It was then combined with the combined set in the same manner and the process repeated for sets four and five. An identical process was then carried out for the second user group.

There was one distinction between the two evaluation processes at this stage. Evaluator One (the experienced evaluator) asked for the interview results to be sorted into tag order, whereas Evaluator Two (the interface designer) preferred them to remain in interview order to provide more contextual information. The total process took approximately one and a half hours for Evaluator One and about three hours for Evaluator Two. Only one comment was made at this stage, in regard to the accounting system, where the experienced evaluator marked two elements as equivalent if 'inputting charges' meant the same as 'invoicing'. From the introduction to the system provided by its designer I determined that this was the case and advised her accordingly.

5.4.4 Comparing the results from the two evaluators

The next stage was to check the two sets of analyses for consistency. Robson (1993, p.338-40) compares the appropriateness of various correlation tests. Pearson's correlation coefficient is based on an assumption of normal distribution which cannot be justified for this data. Other measures are the Spearman rank correlation coefficient and Kendall's rank correlation coefficient (Kendall's Tau). Robson (1993, p.340) states that "Kendall's Tau ... deals with ties more consistently" and must therefore be the most suitable for this data. My analysis followed the step-by-step procedure for calculating Kendall's Tau with ties within conditions given in Robson (1973, p.58-59). The results of the tests are summarised in the following table,

Table 5.7: Kendall's rank correlation coefficient.

User Group 1				User Group 2			
S	*N*	$t_{0.05}$	t_a	*S*	*N*	$t_{0.05}$	t_a
1	28	0.25	0.88	1	37	0.25	0.59
2	22	0.29	0.66	2	65	0.25	0.44
3	19	0.33	0.48	3	58	0.25	0.64
4	37	0.25	0.78	4	19	0.33	0.92
5	31	0.25	0.87	5	33	0.25	0.64

Where:

S is the subject,

N is the number of pairs of ratings,

$\tau_{0.05}$ is the smallest value of τ significant at the 0.05 level for *N*, and

τ_a is the calculated value for the two analyses of the subject's responses.

It can be seen that all values of Tau are well above those necessary to indicate significance at the 0.05 level. The two sets of categories can therefore be regarded as closely equivalent. With regard to the second stage, in which the evaluators looked at equivalences between interviews within a group, comparison is more difficult. Although the categories they were using correlated closely, they were not the same. Direct comparison of the two sets of equivalences is not possible without a common set of categories. It was therefore considered whether the two sets of categories could be merged into one.

Unfortunately, combining the two sets of results would depend on subjective judgement on my part or additional information from the evaluators. Consider, for example, the case where one evaluator tagged consecutive elements '2, 3, 3, 3, 4, 4' and the second tagged the same elements '2, 3, 3, 4, 4, 4'. It could be argued that each has identified the same three meanings in the text, merely disagreeing over the precise point in the sentence in which the signification changed: whether between the fourth and fifth elements or between the third and fourth, leading to a combined record of '2, 3, 3, 3/4, 4, 4'. An alternative explanation is that one evaluator spotted the introduction of a new signification in the fourth element while the other spotted a separate distinction in signification between the fourth and fifth, leading to a combined record of '2, 3, 3, 4, 5, 5'.

Without further information from the evaluators, distinguishing between these cases would depend on intuition or guesswork. The discussion of the results in the next chapter will therefore look at both sets of results, bearing in mind the different backgrounds of the two evaluators.

6 Results

6.1 Quantitative analysis

6.1.1 Presentation of the data

The full responses are listed in Appendix D, split into the response elements used for the content analysis. For ease of reference, each category has been given a two-part number of the form x.y, where x is the category tag allocated by the evaluator and y is the number of the interviewee. Throughout this section and Appendix D, the following abbreviations will be used:

CUG	The Commercial User Group.
AUG	The Academic User Group.
CUx	Commercial User x (where x=1 to 5).
AUx	Academic User x (where x=1 to 5).

As mentioned in the previous chapter, one evaluator was a media studies graduate, the other a psychology graduate working as a researcher in computing. As the background of the evaluators might be relevant when considering distinctions between their results, they will be referred to by following abbreviations:

MSE	Media Studies Evaluator
CSE	Computing Studies Evaluator

6.1.2 Comparing the two sets of evaluations

The table below shows the numbers of response elements and the number of categories allocated by the two evaluators.

Table 6.1: Response categories.

User	Response elements	Number of categories	
		MSE	CSE
CU1	28	14	18
CU2	22	7	12
CU3	20	11	12
CU4	37	10	13
CU5	31	11	18
Mean	**27.6**	**10.6**	**14.6**
SD	**6.88**	**2.68**	**3.13**
AU1	37	13	23
AU2	65	14	27
AU3	58	26	23
AU4	18	11	9
AU5	33	10	20
Mean	**42.2**	**14.8**	**20.4**
SD	**19.15**	**6.46**	**6.84**

The most obvious feature of the table is that the AUG shows a higher number of categories and a much greater standard deviation in the numbers of categories between users. However, the number of subjects is far too small for any difference between the subject groups to be significant. It may also be noted that evaluator CSE identified a larger number of categories than MSE in every case apart from AU4 and AU3. Although this might seem important, the results of the Kendall's Tau test in the previous chapter showed that there was a significantly close correlation between the two categorisations.

6.1.3 Shared categories

A possible reason for the distinction between the two sets of evaluations becomes apparent when the results are compared for the degree to which categories are shared across interviewees. The numbers of unique and shared categories for each interview are shown in the table below.

Table 6.2: Shared categories.

CSE				MSE			
	unique	shared	total		unique	shared	total
CU1	7	11	18	CU1	7	7	14
CU2	4	8	12	CU2	3	4	7
CU3	3	9	12	CU3	8	3	11
CU4	2	11	13	CU4	4	6	10
CU5	11	7	18	CU5	7	4	11
Total	27	46	73	Total	29	24	53
per cent shared			63.01	per cent shared			45.28
AU1	8	15	23	AU1	5	8	13
AU2	10	17	27	AU2	5	9	14
AU3	10	13	23	AU3	20	6	26
AU4	1	8	9	AU4	7	4	11
AU5	11	9	20	AU5	5	5	10
Total	40	62	102	Total	42	32	74
per cent shared			60.78	per cent shared			43.24

It can immediately be seen that both evaluators identified similar numbers of unique categories for each interviewee, totalling 67 for CSE and 71 for MSE. By contrast, the total number of shared categories per interview is 108 for CSE against only 56 for MSE. It is possible that the greater number of categories initially identified by CSE means that each individual category is more basic and therefore more likely to match a simple category from another interviewee. For example, for AU1, CSE allocated "so if you want to make changes" to one category, with "and keep the old changes" to a second category whereas MSE allocated both parts of the phrase to a single category. Where the results of this interview matched those of other interviews, MSE marked them as having one shared category, whereas CSE marked them as two shared categories thus doubling the apparent overlap in content between the interviews.

One question which arises from this is whether, in any sense, one set of results is 'better' than the other. There is no objective reason to suppose this, but CSE's work, background and knowledge of computer systems are closer to those of the interface designers who are the target users for the technique. For this reason, examination of the qualitative data will concentrate on his analysis.

6.1.4 Comparing the user groups

The numbers of users identifying a particular category, as tagged by CSE, are shown in the following graphs. The category tags were allocated by reference to the first interview, adapted to include the second interview and so forth. Ranking categories on this basis would therefore bias them in favour of the ordering by the initial interview in each group. To avoid this, the mean of all the category tags for a given category was calculated and the categories ranked on this basis. For example, if a category was the third category to be introduced by User 1, the fourth by User 2 and the tenth by User 4 then the value used in the ranking would be:

$$(3 + 4 + 10)/3 = 5.7$$

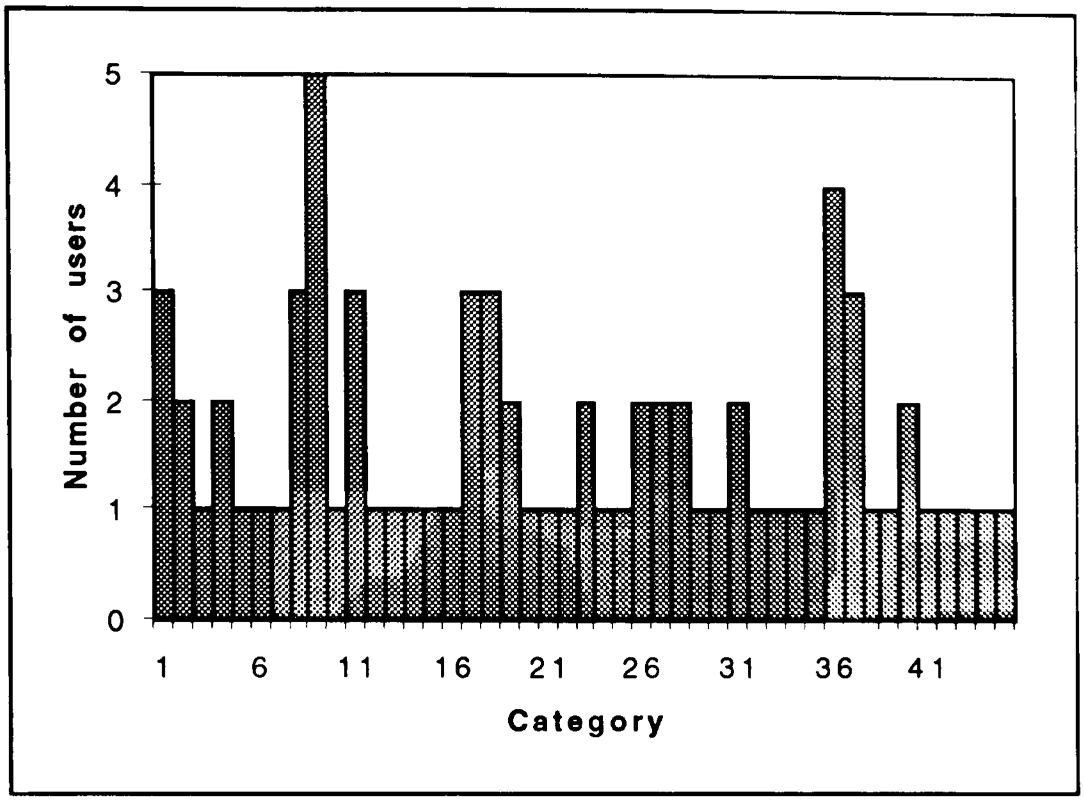


Figure 6.1 CUG: users sharing each category.

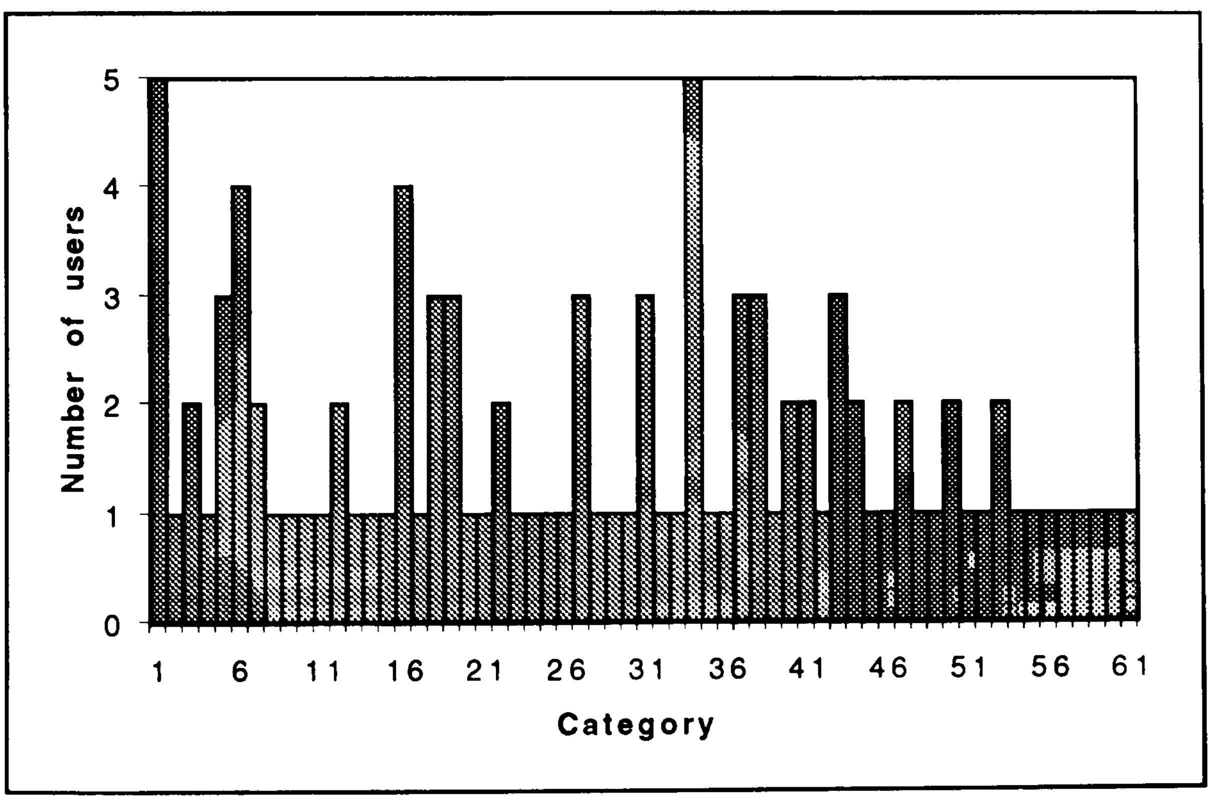


Figure 6.2 AUG: users sharing each category.

Perhaps the most surprising observation is the similarity in form between the two figures. The mean numbers of users per category is similar for the two groups, at 1.62 for CUG, compared to 1.66 for AUG. In both diagrams there are more common categories at the earlier layers of signification but this effect is not very strong – the gradient of the least square line is -0.015

in both cases. Despite this apparent similarity, comparison of the content of the sets of interviews shows some distinct differences which will be discussed below.

6.2 Qualitative analysis

6.2.1 Interview structure

As the table above shows, the CUG interviews uncovered between 12 and 18 layers of signification, while the AUG interviews showed a much greater range, from 9 to 27 layers. However, if AU4 is excluded, the lower figure is 20. Although not the shortest interview, the interview with AU4 was very much shorter than those of the other academic users. He also gave the impression of being somewhat hostile. All other interviewees in both user groups were friendly and interested in the unusual technique.

All interviews followed a similar structure, starting with apparently low level signification and steadily moving towards higher levels with only minor backtracking. Judgement of what consituted a low or a high level of signification was subjective and is only used to illustrate the findings. This will be discussed further in Chapter 7. The interview with CU1 was a typical example, starting, "When you go into that particular screen. You go into 'Navigate' to input an invoice". The user then explicitly backtracked, "That's the first button you press, if you like, before going on to the next fields." CU1 then explained the reason for inputting invoices, "to bill one part of COMPANY to another part of COMPANY," and the reason for doing this, "Because if one part of COMPANY is doing work... then they need to be charged for it." He then moved up to the recording of the information, "So that the books... are... as correct as they can possibly be." His highest level of signification was then given, "It's a requirement under legislation." When asked, "What for?", CU1 ended the interview with, "I don't know." The interviews with AU1 and, to a lesser extent, AU4 showed a minor variation in giving two options (to change the name or to change the format), alternating between higher levels related to the two lower level significations.

All but three of the interviewees ended the interview by saying they could not answer the question, usually with "I don't know". The interviews with CU2, CU4 and AU2 were terminated by the interviewer when the interviewee began to repeat previous answers, slowing down and hesitating as if at a loss as to what to say. For example, CU4 ended by explicitly stating that she was repeating her answers, before returning to her previously stated highest signification, "As I was saying... tasks that I have to do."

6.2.2 The commercial user group

Only one category was mentioned by every member of the CUG: 2.1 "you go into Navigate". This is hardly surprising in that 'Navigate' is the name of the command and the user's first action is to 'go into it'. One category was mentioned by four of the five, 8.1 "to bill one part of COMPANY to another part of COMPANY", the main purpose of the system. A number of categories were mentioned by three of the five users:

4.1 "That's the first button you press..." This represents the most basic interaction with the system, although it is expressed in terms of pressing the metaphorical 'Navigate' button rather than the physical mouse button which is actually pressed.

5.1 "...before going on to the next fields..." The 'Navigate' command is used to select the required screen, the following action being to select the required field for data entry or amendment.

6.1 "To input invoices..."
7.1 "We input invoices..." It is not clear why categories 6.1 and 7.1 were distinguished from one another, perhaps merely because the evaluator failed to spot the common signification. In combination these categories were mentioned by four of the five users. Both describe the main process through which the cross-company billing is enacted.

14.1 "It's a requirement..." The element of compulsion is mentioned by three users, with CU1 seeing the compulsion as coming from Oftel, whilst CU2 and CU4 express it more simply: "I do what I'm told to do," and "Because I have to."

5.2 "So you can navigate around the whole system." This category describes the behaviour of the 'Navigate' command.

For the commercial users, the 'What for?' technique was clearly very successful in disclosing the factors which most analysts would see as fundamental in designing the interface. At a practical level of using the system, the majority of the users mentioned the type of interaction (pressing a button), the command chosen (Navigate), the behaviour of the command (moving around the system), the reason why it is necessary to move around the system (to get to the right screen), the reason why they would want to get to a particular screen (to select a field for entry or amendment). A majority of the users also expressed an awareness of the role of the system in the business (cross-company billing) and how this is achieved (invoicing).

In contrast with this level of agreement, the interviews uncovered some interesting distinctions in higher level signification which might give concern. The actual reason why the system was built – "a requirement under legislation and under Oftel that each part of COMPANY should be responsible for their own accounting purposes" – was only mentioned by CU1, the most senior member of the team who supervises the others. I verified that this was the case with the systems analysts who had specified the system. The only other interviewee to express a higher level of signification related to the company was CU5, who justified the system incorrectly: "To provide information to shareholders. To see how good we are doing as a business."

It is widely recognised that understanding the purpose of a task is an important factor in employee motivation. Huczynski and Buchanan (1991, p.73) identify "experienced meaningfulness" as one of the "three psychological states critical to high work motivation, job satisfaction and performance," defining this state as "the extent to which the individual considers the work to be meaningful, valuable and worthwhile." For an employee to consider his or her work to have a worthwhile purpose obviously depends on the employee knowing what that purpose is. This knowledge is also likely to contribute to the third of the three critical states: knowledge of results.

6.2.3 The academic user group

For the academic users many more categories were shared by a majority of the users, with two categories mentioned by all five interviewees:

1.1 "It's for saving the contents of a file..."

8.1 "...if you've got several different versions."

1.1 is a simple statement of the functionality of the command, while 8.1 is one of the possible reasons for using the command. Two categories were raised by four of the five users:

5.1 "And one to change the type of file..."

13.1 "...other kinds of software, other packages..."

Again, these describe another possible reason for using the command. A large number of categories were mentioned by three of the five users:

4.1 "One to change the name to a different name."

9.1 "...of the same file."

16.1 "...you'd want to save..."

18.1 "...so if you want to make changes..."

19.1 "...and keep the old changes..."

21.1 "Historical record."

8.2 "For using it, I mean."

23.2 "...or whatever the purpose of the file is."

Some of these categories deserve examination, such as 4.1 and 5.1 which are not technically accurate. The 'Save as...' command does not change the name or type of a file. It saves a copy of the file under a new name or type; unless overwritten, the original file remains on disc with its original name and type. One possible reason for this will be discussed below.

6.3 Assessing the value of the technique

6.3.1 Duration of interviews

One important factor to consider in assessing the practicality of the technique for potential users is the time taken for an interview. Times for each individual interview are shown in the table below.

Table 6.3: Interview duration.

Commercial User Group		Academic User Group	
Interviewee	**Duration**	**Interviewee**	**Duration**
CU1	1 min 13 sec	**AU1**	1 min 49 sec
CU2	0 min 43 sec	**AU2**	3 min 37 sec
CU3	1 min 20 sec	**AU3**	2 min 35 sec
CU4	1 min 54 sec	**AU4**	0 min 58 sec
CU5	2 min 1 sec	**AU5**	1 min 26 sec
Mean	1 min 26 sec	**Mean**	2 min 5 sec

On average, the interviews took less than two minutes each, the longest being under four minutes. One potential criticism of the interview technique might have been that, when probed in this way, users would reflect on the interface, inventing layers of signification to please the interviewer. It is clear, however, that the users had little chance to invent significations in this way and that the answers were given with little time for thought. As for the practicality of the technique for use by designers, even with pre-interview set-up, post-interview explanations and occasionally waiting for a user to be free, each group of interviews was completed in less than an hour. In practice this would make no significant difference to the total time taken for effective user requirements gathering and would be negligible in the overall development life cycle.

6.3.2 Metaphor issues

The 'save' metaphor is now dead for many users, certainly for experienced computing postgraduates such as AUG, and is included as a computing term in recent dictionaries such as Chambers (Schwarz 1988) which lists one meaning as "to store (data) on a tape or disc." It should be noted that this does not preclude the possibility of problems based on its metaphoric origins; as explained below, 'file' is also a dead metaphor but still caused confusion among users. However, the 'save' metaphor appeared to cause no problems among the users, with all members of AUG making statements in line with the dictionary definition as their first significations, such as 1.1: "It's for saving the contents of a file."

For the general population, the 'navigate' command is still a live metaphor. Chambers (Schwarz 1988) does not list it as a computing term and defines the verb as, "to direct the course of: to sail, fly, etc., over, on or through." Whether the metaphor remains alive for the users in this study cannot be answered from the interview results alone, but CUG were generally less familiar with computers than AUG and it seems likely. The general intention of the metaphor is that users should use the facility to steer their way around a system like steering a ship across an ocean, as indicated by the ship's wheel icon used to represent Netscape Navigator 4.0 (Netscape 1998).

CU1 did not mention the functionality of the command apart from the statement, "You go into 'Navigate' to input an invoice," giving no indication that he necessarily understood the meaning of the command. CU2, CU3 and CU4 all referred to the metaphorical meaning in their replies:

CU2 1.2- 7.2 Just press the 'Navigate' and then it takes you to all the other bits. So you can navigate around the system. So it'll get you to different screens to do your work.

CU3 1.3, 2.3 Just to show the system where to go.

CU4 4.4 It takes you on to another route.

The only questionable aspect of this understanding lies in way that both CU2 and CU4 used the phrase, "it takes you," as if they regard the system as being in control rather than themselves. The least experienced of the users, CU5, re-interpreted the meaning of the command as "Find, I'd assume find." Generally, however, the metaphor appeared to work effectively and none of the users apart from CU5 appeared to have any problems in understanding how to use it.

A more important metaphor issue to be considered is the users' understanding of the 'file' metaphor by AUG. Problems with this metaphor could explain the 'false' significations which all of the academic users gave. This could come from confusion as to what a file is, whether an identical copy made to another drive (3.3) is still the same file, whether a file remains the 'same' file if it changes its name (4.1) or its file type (5.1), or if its contents are changed (18.1). Although 'file' was not the term under consideration, the 'Save as...' command is an item on the 'File' menu and the object to be saved can only be a computer file. The 'file' metaphor was used by all users in the AUG, with 28 references to 'files', and only one to 'document'. In order to examine this it is necessary to look at the historical origins of the 'file' metaphor.

Before the widespread use of personal computers, the term 'file' was applied to a collection of computer data metaphorically equivalent to the physical file it replaced. For example, Kilgannon (1980, p. 97) describes equivalent processing methods a systems analyst should consider: manual processing, office machine processing, unit record processing and computer processing. He then goes on to examine manual processing as the first option to be considered, introducing the following list of files and filing equipment.

Table 6.4: Filing equipment(from Kilgannon 1980, p.99).

Document filing	Book binders (ring or post) Box files Drawer cabinets –folders –suspended folders Lateral filing units Fire resistant cabinets Safes
Card filing	Plain boxes (blind filing) Rotary boxes Card wheels Visible edge card trays (Kardex) Vertical ledger card visible records
Plan filing	Drawer filing Vertical plan files Lateral plan files
Strip index	Book binders Trays Rotary trays
Wall charts	Plan boards – visible, magnetic Peg boards Manual entry
Manual punched cards	Edge punched Body punched

At this time computer files were seen as the metaphorical equivalent of the physically organised data within an office environment. The ground for this metaphor lies in the structured organisation of the data so that, for example, typed documents are not listed as a type of file. This structured organisation is crucial to the metaphor, as shown by the following diagram in which Kilgannon lists the various data structures employed in computing.

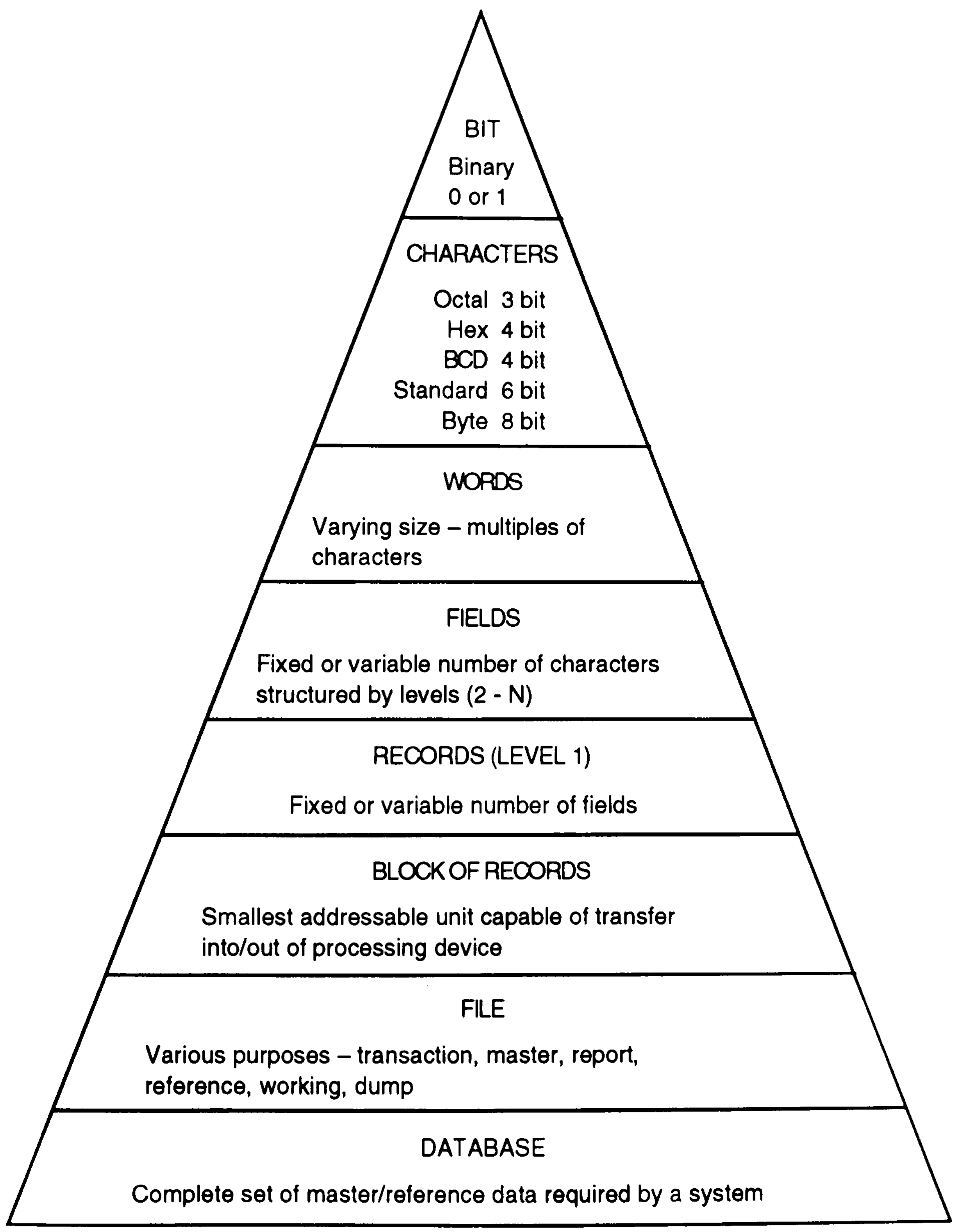


Figure 6.3: Data structures (from Kilgannon 1980, p.59).

In the context of the other data structures, it is relatively easy to determine what a file is. A physical file has a specific location, if not a unique name, whilst a file that forms part of a mainframe data processing system will have a defined internal structure and a specific name. With the move to personal computing, the same term has been generally applied to the 'files' used on PCs, even though these do not share the formal characteristics of the physical files or mainframe computing files. Some interfaces based on the desktop metaphor have attempted to apply the more appropriate metaphor of 'document' but this has not been done consistently. For example, although the Lisa and Macintosh use the terms 'document' and 'item' to describe entities on the desktop, the 'File' menu forms a standard element of the Apple Human Interface Guidelines for applications (Apple 1987). Similarly, Microsoft Word, the application in this study, employs the standard 'File' menu but refers to the file as a 'document' in other places such as the 'Format' menu.

Wordprocessing files do not have any visible structure. Users are free to structure their documents in any manner they care to, backing up or transferring data to other programs according to whim, with no formal control. Thus, whereas mainframe control procedures would regularise back-up procedures with clear unambiguous names for back-ups or historical records, consistent with the original metaphor, personal back-ups are likely to be ad hoc. Users referred to different versions of the same file, copies of the file and back-ups without any consistency.

6.3.3 Higher levels of signification

Assessing the highest level of signification for each user cannot be entirely objective but it is generally quite clear that when a respondent has answered that A is done because of B, then B represents a higher level of signification. There is not necessarily a single level of signification at the highest level and the following list combines some categories which formed part of a single statement. The list shows the highest levels of signification for the commercial users, with the number of other users mentioning the same categories in brackets.

CU1 14.1/25.1 It's a requirement/ under legislation. (2/0)

CU2 10.2/11.2 Because I need the money/ and I do what I'm told to do. (0/2)

CU3 13.3/14.3 To get to the screen/ I want to go to. (1/1)

CU4 14.4 Those are the tasks/ that I have to do. (1/2)

CU5 19.5/20.5 To see how good we're doing as a business/ and we're making a profit. (1/0)

It can be seen that there was a small degree of shared signification amongst the higher levels mentioned by the commercial users, with each of the issues being at least partially mentioned by other users. As discussed above, two of them saw the highest level of signification in company terms, one incorrectly. CU2 and CU4 both saw the highest level of signification in personal terms, doing what one is told to do, whilst CU3 failed to mention any signification above the level of the operation of the system.

By contrast with the commercial users, the highest levels for the academic users were all different and all related to their own areas of work:

AU1 22.1/23.1 So that you can keep your train of thought - what changes have been made. (0/0)

AU2 24.2/25.2/11.2 To make it easier for other people/ to run it/ or use it. (1/0/0)

AU3 28.3/29.3/30.3/31.3 To send these letters/ to my contacts. To receive a response from them/ if it's an invitation./ They'll contact me. (1/0/0/0)

AU4 7.4/8.4 Because it's a slightly different application you're developing/ or a different direction. (3/4)

AU5 9.5/10.5 Probably you want to continue working/ somewhere else. (0/0)

AU1 and AU5 gave totally unique significations, while AU2 and AU3's highest levels were only partially mentioned by a single other user each. However, the first of AU4's categories was mentioned by a total of four users and the second part by all five. The interview with AU4 was much shorter than the interviews with the other academic users and his highest level of signification was not comparable to the others. He failed to give any non-generalised significations, cutting the interview short with the complaint, "You're getting too wide." All other users went further on these points, elaborating on the reasons why they might wish to develop a new application or go in a different direction.

The split within the CUG between users who related the highest level of signification to their own purposes and those who related it to the business did not occur in the academic group. This is not surprising in that doctoral studies are largely self-motivated, although one student did refer to saving data in a different format "to make it easier for other people to run it or use it." The commercial system was designed for use by a specific group of users to meet a specific business function, whereas wordprocessors are designed to be used by anyone who has access to a PC, leading to the wide range of purposes that the academic users identified.

The question of whether the 'What for?' technique is of value to system developers and designers depends to a considerable degree on whether the significations uncovered are relevant to the design. In the case of the commercial users, as discussed above, the importance of the higher level significations lies more in management issues than in design issues. The interviews also revealed whether work is self-motivated or whether users are simply doing it because they "need the money." The significations uncovered among the academic users would be of more direct usefulness, showing a range of different purposes some of which might well be beyond the designers' original scenarios for possible use of the system. This will be discussed in Chapter 7.

6.3.4 Conclusions

The 'What for?' interviews were intended to test the last two assertions listed at the end of chapter 3:

Assertion 4	The recursive nature of signification leads to a structured model of multiple layers of signification.
Assertion 5	Further layers of signification can be uncovered by asking of each layer, "What does that signify?" or, more simple, "What for?"

The Layers of Signification (LoS) model was extensively described in chapter 3. The structure of the interviews showed considerable concordance with the LoS model in that each respondent began with lower levels of meaning, progressing to higher and higher levels. The interviews cannot prove that the LoS model lies at the centre of human thought but they do affirm that it is a valid way to look at the meaning of an interface to a user, confirming Assertion 4. The 'What for' technique was successfully used to uncover the layers in an efficient manner, confirming Assertion 5. A final question was also posed:

Question 5	What is the highest level of signification that matters to the user?

The highest level of signification revealed by each user was listed in section 6.3.2. These were not necessarily the highest levels that existed for each user but can be considered to be the highest level that mattered to the user and was considered worth mentioning at that time. Two interviewees stopped at a comparatively low level and showed an unwillingness to reveal higher levels. This cannot be taken as indicating that higher levels did not exist and may well have been caused by hostility towards the interview method or embarrassment at revealing more personal aspects of the interface element's meaning.

The types of high level signification which were revealed included aspects which are rarely taken into account by interface designers, such as the users' motivation for doing their work. Whether designers should consider these factors will be further discussed in the next chapter.

7. Conclusions

7.1 Summary of findings

Although eminent writers such as Kay and Nelson (see Chapter 2) have attacked the use of metaphors in human-computer interfaces, this thesis has demonstrated that they are impossible to avoid. Indeed, some studies reported in Chapter 2 suggest that metaphor is more than a central feature of communication and may be fundamental to human thought. Attacks on the use of metaphor appear to be based on particular examples of bad metaphors or employ a limited view of what a metaphor is:

> Here is the problem with metaphors: you want to be able to design things that are *not* like physical objects, and the details of whose behaviour may float free, not being tied to the details of some introductory model. Metaphors are like WYSIWYG: useful in limited contexts, but ultimately a drag, a dead anchor. (Nelson 1990, p.237)

Metaphors do not have to be based on physical objects, as shown by examples such as 'demons' and 'wizards', but even where they are, their functionality can be transformed through 'magic'. There is also the question of what to use instead of a metaphor; as I showed in Chapter 2, Nelson's suggested replacements for current interface metaphors are themselves metaphors, though perhaps better ones. Earlier literature, particularly the work of Carroll described in Chapter 2, showed that metaphor can act as a powerful mechanism for learning, but metaphors can also interfere with the learning process leading to problems for the user. The aim of this thesis has therefore been to accept that interface metaphors exist, try to understand their role in HCI, and find ways in which to assess the suitability of an interface metaphor in a given situation. If metaphor is central to thought then the limitations of metaphors are ultimately the limits of human thought.

In a discussion of metaphor in film, Whittock (1990) suggests that the study of metaphor could proceed in two directions, towards cognitive psychology or towards "rhetoric and strategies of communication." Previous studies of metaphor in the human-computer interface have concentrated on the first of these approaches; I decided to examine the second. The rhetorical approach is descriptive, classifying the devices used in communication, with particular concentration on non-literal devices, such as metaphor, which are known as 'tropes.' This descriptive approach was applied in a trope analysis, described in Chapter 4, which showed the widespread use of metaphor and metonymy in MS-DOS, even though this is not generally seen as a metaphor-based interface.

Modern studies of metaphor have developed beyond rhetorical classification towards more complex models of communication, a field known as semiotics. Like rhetoric, semiotics sees metaphor as one of many tropes available for our communication, although most writers recognise it as one of two master tropes, the other being metonymy. Metaphor, in particular, is seen as central to the development of language and that of comparable semiotic systems. Following this direction, it becomes impossible to study metaphor as an element in its own right; it must be considered as one part of the semiotic process. Conversely, any principles or techniques which help in understanding the semiotic process will be of particular help in understanding metaphor.

A semiotic model was developed in Chapter 3, based on studies by de Saussure and Barthes, together with Eco's Theory of Signification. Other researchers have used Eco's Theory of Sign Production to examine the way in which a designer expresses concepts through the interface. This thesis concentrates on the perception of the interface by the user, a process known in semiotics as signification. Signification is a recursive process leading to many layers of meaning inherent in even the simplest interface element. These layers of signification (LoS) are not alternative meanings; they co-exist in the user's mind, affecting the way in which the user will approach an interface or a command and the way it is used.

Having proposed the LoS model as a semiotic approach to metaphor, a number of questions and assertions were put forward. It was important to compare the semiotic approach with that of cognitive psychology. One previous experiment had used an approach based on mental models to compare three alternative interface metaphors with very positive results. However, the original experiment was carried out with computing postgraduates and compared three systems with extremely limited functionality. As described in Sections 4.4-4.8, the experiment was extended to a more realistic manufacturing scenario, with the users needing to carry out three distinct functions to complete the scenario. This experiment showed none of the effects shown in the first experiment, with users failing to form accurate or consistent mental models. Even if the mental model approach was correct, it was too simplistic to account fully for user behaviour.

Three metaphors were used in the experiment, chosen to represent three distinct types of metaphor – spatial, activity-based and interactional – which had previously been identified as distinct categories. An analysis of the subjects' responses showed that they did describe the experiment within the terms of the chosen metaphor category. More importantly, inspection of their descriptions showed that they described the system in terms of different layers of signification; the spatial system in terms of its interface, the activity-based system in terms of its functionality, and the interactional system in terms of the tasks the system could support.

The LoS model was used to successfully carry out a simple semiotic analysis of the Macintosh user interface, described in Section 4.3. Having confirmed its applicability to the task, the 'What for?' technique was developed as a method for designers. A study was carried out with users of two existing systems, one group using a generic application, the other a bespoke system. The technique provided substantial information in a short time, showing its practicality for use by system designers. Content analysis of the results showed that users identified between nine and 27 distinct layers of signification. The potential usefulness of these will be discussed below, as will the potential value of the technique in choosing appropriate metaphors.

7.2 The LoS model

7.2.1 Multiple signifiers

Examination of semiotics has shown that the role of metaphor can be seen as part of a wide range of signification of the user interface, in which the interface signifies many different things to the user simultaneously. To reduce the study of human-computer interaction to a single level is likely to over-simplify this very complex process. Other researchers have also suggested more complex structures to describe human-computer interaction, though not in terms of signification. It is therefore necessary to explain what the concept of signification is, and how it differs from other approaches which might seem similar on the surface.

The most important factor in looking at signification is that all the factors concerned are in the user's mind. The *signifier* is not the physical *sign* but the observer's most basic interpretation of the sign. There are many signs with multiple meanings, any of which could provide a different signifier. For example, in this thesis, the initials 'PC' would be taken to refer to a personal computer; in other circumstances they could refer to a police constable or to something being 'politically correct'. These not only lead to different significations but actually represent different signifiers. In this thesis, the full term 'personal computer' could have been used to generate a similar signifier to the term 'PC'. However, the exact nature of the sign will always influence the signifier. Although the signifiers for 'PC' and 'personal computer' are similar, they are not identical: use of the term 'PC' might be taken by the user to refer only to IBM PC compatible machines, whereas the same user might see 'personal computer' as a more general term, including other types of microcomputer. The signifier is thus a product of the specific sign and its context.

Signification is the operation which takes place once this initial observation has been made. For example, to some people 'politically correct' might be a term of praise, whereas others might see it as a term of contempt, depending on their personal political views. Occasionally, deliberate puns or accidental confusion might also present the user with two or more signifiers at the same time. Many graphical examples can be found within books on the psychology of perception, such as the widely reproduced figure below, which can be interpreted as a vase (white area) or two faces in profile (shaded area).

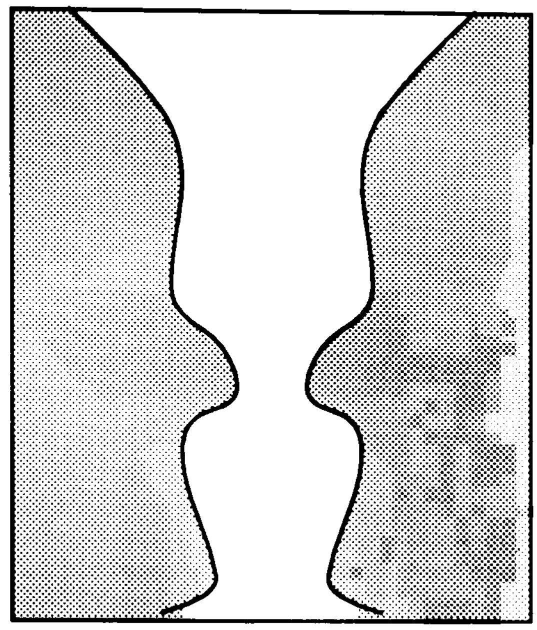


Figure 7.1: Example of dual signifiers from a single sign.

Obviously ambiguous signs with multiple signifiers might sometimes occur in user interfaces, particularly with simple icons. However, these do not need any further consideration as the ambiguity will rapidly become apparent when examining the higher layers of signification they lead to. What is more important is the fact that a single signifier can be subject to many different acts of signification, leading to many potential signifieds, whether to different users or as multiple signification for an individual user. To use the alternative terminology mentioned in Chapter 3, an ambiguous sign might be seen as having two or more *denotations,* all leading to different *connotations.*

7.2.2 The conceptual space uncovered by LoS

Some discussion, particularly in section 6.2.1, has considered how 'high' a level of signification is. This observation has been based on the use of the term 'high' in general conversation, such as reference to 'higher level motives', and consideration of how closely the signification accords with the immediate use of the interface element (a close accordance being seen as 'lower' than a more general observation). The observations made were subjective and should not be controversial. For example it would be difficult to argue that moving to a different screen represents a higher level of signification than consideration of a company's legal duties.

Care should be taken not to extend this concept too far. Although there is a tendency for users to begin with obviously low level significations and move on to higher ones, there were many examples of backtracking and occasional jumps to alternative significations at apparently very different levels. The process might be seen as analogous to someone starting at a particular tree at the centre of a wood and then wandering around, exploring the other trees. Obviously the perimeter of the wood will take some time to reach and will not be among the first places explored. It is also unlikely that the person will move in a straight line from the centre to the edge. Nearer spots might be overlooked initially and returned to later on; interesting areas might be re-visited on purpose. Also, it is extremely unlikely that the person will visit every single tree unless the wood is deliberately surveyed and mapped out.

Presenting this graphically, we might see a pair of interviews in the following manner:

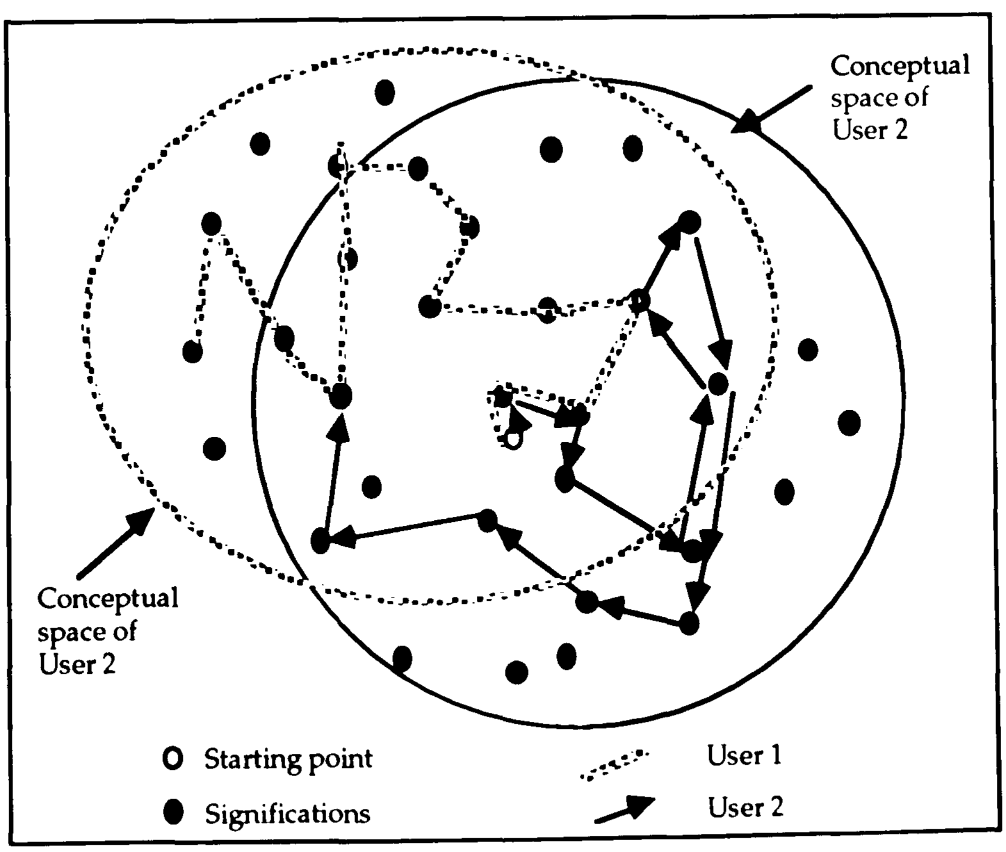


Figure 7.2: Exploration of conceptual space.

Figure 7.2 shows some of the potential significations for two users and the way in which their interviews might uncover some of them. LoS interviews are undirected and allow the user to 'wander around' their conception of the interface element, moving from one signification to another. Of course, some of the significations will be perceived by the user as more important than others and it is likely that the user will feel drawn towards these, but there is no guarantee that the user will necessarily mention all of them. Repetition of the method with larger numbers of users will obviously help to ensure that lower level significations (those close to starting point) will be uncovered but the method can never be exhaustive, as there is no absolute boundary on the potential conceptual spaces different users might have.

7.2.3 Contrasting LoS and GOMS

Some aspects of the LoS model are superficially similar to the GOMS model developed by Card et al (1980a). GOMS has formed the basis for many variations and refinements (John 1996) but these do not change the underlying model. This model is based on Goals, Operations, Methods and Selection rules, and depends on breaking a task down into these components to the lowest level, the operations, which represent simple keyboard activities or mouse actions which can be timed. The overall time taken to complete the task can then be worked out, giving a method for comparing the efficiency of different interfaces.

Examination of the results of the 'What for?' interviews in the previous chapter shows that many responses could be categorised as GOMS entities. For example, examination of CU1's responses reveals goals such as, "to bill one part of COMPANY to another part of COMPANY". This goal is broken down into methods, "That's the first button you press, if you like, before going on to the next fields to input invoices.". The methods are also broken down into a sequence of operations, such as "That's the first button you press". Although CU1 did not express any selection rules, other interviewees did, such as AU1, who described his selection criteria for choosing whether to change the name or the type for a file being saved.

There are two very distinct differences between LoS and GOMS. Firstly, GOMS is a top-down approach, taking as given that a user wishes to complete a particular task and that the task is well enough understood for the user to break it down into the components required for its successful completion. By contrast, LoS is unstructured, allowing users to express their own beliefs about the interface and the task. The second difference between the approaches is the prescriptive nature of GOMS which only deals with specific types of entity. The LoS model depends on the signification to the individual user. If a user thinks that the goal is important then he or she will mention the goal; in other cases the sequence of operations or the selection method might be seen as more important and the goal not mentioned.

The first difference between the approaches can be summed up in a simple diagram:

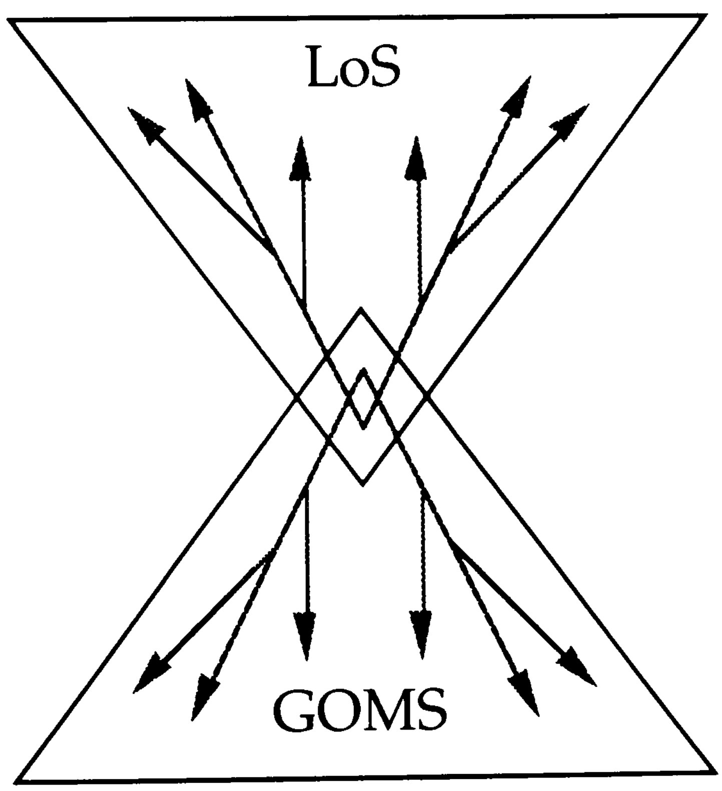


Figure 7.3: Comparison of LoS and GOMS.

From this it can be seen that the two approaches could be seen as complementary and could be usefully employed in conjunction. An example of the use of GOMS is given by Card et al (1980b) in which it was used to analyse a manuscript editing task. This bears comparison with the 'What for?' interviews with AUG as the users in both experiments were using a text-editing application. As the interviews showed, users were using Microsoft Word for a wide variety of reasons not normally associated with text-editing or word-processing, such as conversion of file types for use by other programs, or construction of personal databases. Use of the LoS model on its own can give no information about the efficiency of a particular interface, though it might indicate possible problems; use of the GOMS model by itself might be seriously flawed if the tasks chosen for analysis are based on a misunderstanding of what the users use the application for or why they want to carry out particular tasks. A designer could begin with 'What for?' interviews to develop a fuller understanding of the purposes to which the system is put and the tasks likely to result from them, followed by a GOMS analysis of those tasks to compare the interfaces.

The second distinction between the two methods is perhaps more important. By choosing pre-arranged types of entity, the GOMS analysis takes no account of what is important to the user. Card et al are clear that their text-editing example looks at a *routine cognitive skill* (Card 1980b p.33 original italics). Many users carry out routine computing tasks, such as CUG who regularly input invoice data on the same system. Many other users, such as AUG, use computer applications for their own purposes. In general, the move from mainframe computing towards personal computing has been a move away from routine use of computers towards adaptive use of computers to meet personal goals. In these circumstances, it is essential to develop models such as LoS which allow the users to express both the differences and the similarities in use amongst themselves in their own terms. The LoS model and the 'What for?' method represent a very efficient approach to gaining a substantial part of this information and may even uncover levels of signification not normally considered in GOMS which could help to provide a wider context for the user's behaviour and motivation. However, the LoS model can never provide a substitute for GOMS; although goals, operations, etc., might be uncovered, LoS is non-exhaustive and important GOMS entities might be entirely overlooked.

7.2.4 The nature of signification

Although uncovering layers of signification can lead to the expression of goals or operations, they do not represent a chain of causality. A signified, *x*, might signify *y* to a person because of signification *z*, but this is not necessarily the signification the person will be conscious of. Nor does the LoS model represent a chain of processes (as one might uncover in forming a process flow chart), nor a logical chain. Each separate layer of signification might be formed in a different way, one signification representing a goal, the next layer a causal chain, the next an operational sequence. We are not usually aware of why signification takes place and might never know. For example, people with phobias may have no knowledge of the reason that a particular thing signifies fear to them.

Some of the responses obtained from a user could be post hoc justifications for their actions or their perception. The recursive nature of the LoS is such that the false signification could lead to further levels of erroneous signification. For example, the person who is frightened of spiders might justify this by saying that it is because the spider has eight legs. Although this might be false, the person may then attach the signification of fear to another eight-legged creature, such as an octopus which had never previously been seen as an object of fear. Thus, although some significations are 'false' in one sense, they are true to the user and cannot be ignored by the designer. In some cases, the false signification might appear to lie outside the scope of the interface designer but there will often be something that can be done to deal with the problem. For example, CU5's incorrect justification for the accounting procedures might prompt the designer to change the name of the screen or the system to 'OFTEL Accounting' to make its true purpose clear. This might lead to a better understanding on the user's part but obviously needs to be considered against other constraints.

7.2.5 The higher levels of signification

One of the most notable features of the LoS model and the results of the 'What for?' interviews is the fact that it includes very high levels of signification well outside the factors usually considered by a system or interface designer. A designer might therefore dismiss the approach as irrelevant to the practical issues involved in design. However, to deliberately confine the scope of the model to those factors which the designer knows are important would undoubtedly lead to the model missing out some factors which could be of importance for a particular interface. For example, an interface to a one-off standalone system might require consideration of a manual system it will replace; another might not replace a manual system but could form part of an existing suite of programs with which its interface must be integrated. An effective model must be capable of taking any factors into account, including factors which are currently unforeseen, even if this means the inclusion of irrelevant factors which can then be discarded.

Although not constrained in this manner, the LoS model does have constraints in that it is limited to factors which matter to the user. Although the designer might dismiss certain factors as beyond the scope of the interface design, the fact that the user raises a point is enough to suggest that the designer should at least consider it. The converse does not apply, in that many important factors will not be raised by the users. The LoS model should therefore always be used as one of a number of tools to be applied by the analyst or the designer, as with the example of the relationship between LoS and GOMS outlined above.

7.2.6 Practical use of the LoS model

As previously stated, it is intended that the approach, particularly use of the 'What for?' technique, should be of practical use to interface designers. In practice, this will mean the simplification of the method to provide 'quick-and-dirty' versions. However, there are limits to how 'quick-and-dirty' any method can be whilst still giving valuable results. The degree of concordance between the interviewees in the two user groups shows that a sample of five users is sufficient to yield useful information. However, it is unlikely that the sample could be much smaller, as each group included one interviewee who ended the interview at a comparatively low level of signification.

One factor which could obviously be excluded from practical use of the technique is the content analysis, which took much longer than the interviews themselves. An analyst or designer could carry out a rough analysis of the interviews, particularly if experienced in the task, but most of the important information should be immediately evident from a simple examination of the users' responses. The most extreme reduction in the method would be for the designer to carry out an examination of the signification to him or herself in the manner of the analysis of the Macintosh interface described in Section 4.3. Such an analysis would obviously be highly biased by the fact that the designer would know the purpose of the interface but could provide a 'first pass' to design a prototype to be used in the user interviews.

7.3 The LoS model and interface metaphors

This thesis began by considering metaphor in the human interface but may have appeared to move far from this root. Looking at the role of metaphor in general led to consideration of tropes and the semiotic model of communication. Metaphor is a central feature of language and other semiotic systems, as was explained in Chapter 2, and any examination of metaphor depends on a model of semiotic processes such as the LoS. It is only now that the model has been developed and assessed in relation to computer interfaces that its role in explaining the power of interface metaphors can be summarised.

7.3.1 Identifying inappropriate metaphors

Placing interface metaphors within the context of the LoS model, it is immediately apparent that both the tenor (the interface element) and the vehicle (the concept used to form the metaphor) will have their own layers of signification. Consider some of the problems with interface metaphors discussed in Chapter 2, such as the findings of Carroll and Mack (1984) that the misuse of a text editor was due to the users' adoption of a typewriter metaphor, or the examples of 'male' and 'rape' metaphors from Grundy (1996). The typewriter/text editor problems could be reduced to a simple functional mis-match; the examples from Grundy could not possibly be explained in this way. Both examples, however, make immediate sense when viewed in the light of the LoS model.

With the typewriter/text editor example, the layers of signification for the user are likely to be very similar, although exhibiting a mis-match at some of the lower, functional levels of signification as shown in Figure 7.4:

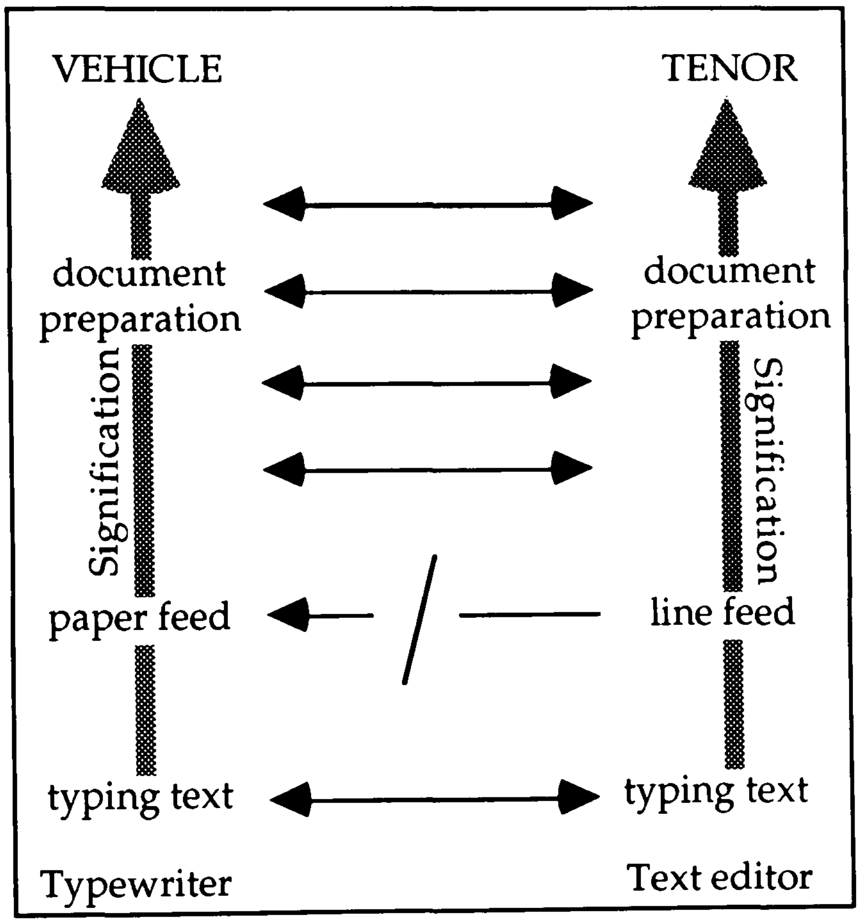


Figure 7.4: LoS mismatch, typewriter and text editor.

By contrast, the match between the layers of signification for 'abort' and the actual action (designated here as 'abandon') is only successful at the very lowest level, with higher levels being radically different, as shown in Figure 7.5. What these higher levels are will obviously depend on the individual, but for 'abort' these might include moral and religious issues, concerns about women's rights, men's roles, the definition of a human life, and more, none of which would have any relevance to the computing command. In this context, Grundy's extreme reaction to 'rape metaphors' makes perfect sense.

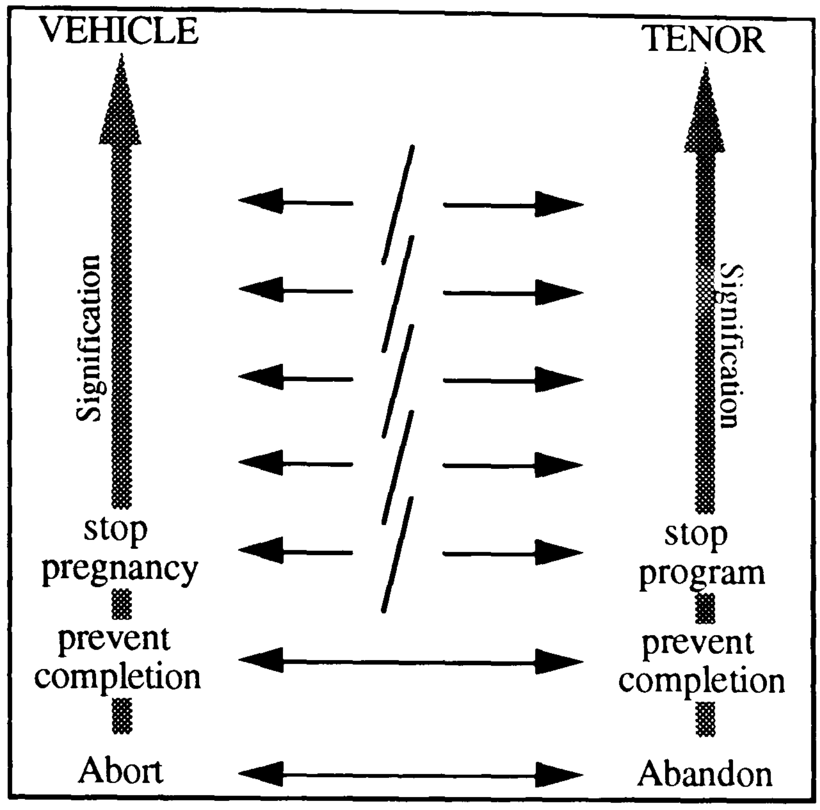


Figure 7.5: LoS mis-match, abort and abandon.

When considering a potential interface metaphor vehicle, many of these layers of signification would be readily apparent to the designer. A simple self-directed semiotic analysis of the type carried out with the Macintosh interface in Chapter 4 would readily demonstrate that the term 'abort' might not be the most appropriate metaphor to choose. However, the designer might come from a background in which the term is more likely to apply to the abandonment of a missile launch than a pregnancy. A more balanced assessment would therefore depend on uncovering the signification of the potential metaphor vehicle through 'What for?' interviews with a range of potential users.

7.3.2 Ground and tension

The examples examined in the previous section could also be interpreted in more traditional terms of ground and tension. It could be argued that the mis-match in the 'abort' example comes from the tenor and vehicle having too small a ground and too great a tension, whereas the typewriter and text editor share a greater ground leading to less tension. However, this does not correspond to the traditional definition of the ground as the features common to the tenor and the vehicle. The literal meanings of abort and abandon are extremely similar. Chambers dictionary (Schwarz 1988) gives the definition of abort, when applied to a mission, as "to check or call off at an early stage" whereas that of abandon is "to give up". When applied to abandoning or aborting a computer command, either meaning is literally true - there is not even a metaphor, let alone any tension. However, this only applies to the literal meaning, or denotation, of the term. The mis-match in using 'abort' becomes apparent when examining the connotation of the term.

As stated in chapters 2 and elsewhere, most researchers have seen the value of metaphor in computing as introducing a familiar concept to aid the user in learning a new concept. Obviously this is only possible when the vehicle and tenor have a considerable ground in common. This can be extended further towards what might be termed the 'connotational ground'. The 'What for?' interviews provide a potential method which can be used to compare the signification of a potential metaphor vehicle to the user with the signification to the designer of the intended function or interface component. Judgement of what constitutes the connotational ground will obviously be subjective but significant mis-match will be easily apparent.

Although the designer should be aware of any mis-match between the significations of the tenor and the vehicle, this does not necessarily invalidate the metaphor. Chapter 2 looked at the concept of magic when implementing metaphor-based features. As Kay (1990, p.9) put it, "it is the magic – understandable magic – that really counts... that must be most strongly attended to in the user interface design." When considering this in relation to the LoS model, the critical term is 'understandable'. Understanding can come from explicit training or explanation to the user but should ideally be rooted in the signification.

Consider, for example, the magical features of the Macintosh folder listed by Dieberger (1994a, p.60):

- It never can get "full" as long as there is space left on the hard disk.
- It can contain other folders.
- "Things" in the folder are sorted automatically in the list representation of its contents.
- It is possible to search the folder without opening it.

If questioned through a 'What for?' interview, it is likely that most users would mention 'putting things in' a folder as one level of signification. The first two magical features listed by Dieberger extend from this signification and do not conflict with it in the way that the paper feed/ line feed significations conflicted. The contents of physical folders are often sorted into order, for example according to date and it is quite likely that this signification will also occur to users. Extending this signification to automatic sorting is simple and unlikely to cause problems. The fourth feature, searching the contents of a closed folder, does appear to represent a more important mis-match. However, this feature is not actually part of the 'folder' metaphor; it forms part of the 'Find file' command. Options in the 'Find file' dialogue include 'version', 'lock attribute' and other technical terms which owe nothing to the folder metaphor. The mis-match here is not between the signification of the tenor and the vehicle but between the desktop metaphor and the strictly functional 'Find file' command.

In summary, successful 'magic' is likely to come from the designer examining and extending the signification of the metaphor vehicle to the user rather than explicitly denying or ignoring it. For example, it might have been useful to take the chapters of this thesis, each of which is a separate document, place them into a folder and then turn that folder into a single document. This would undoubtedly conflict with the signification of a folder (and that of a document) to most people: in Kay's terms it would be magic, but not 'understandable magic'. If, however, 'chapter' and 'book' metaphors were used instead of 'document' and 'folder', then the operation becomes quite understandable.

7.3.3 Finding appropriate metaphors

Unlike the CUG responses, the AUG responses showed a multiplicity of signification at similar levels. Users talked of using the 'Save as...' command to make an identical copy of a file to another drive (3.3), to change its name (4.1) or its file type (5.1), or to save changes to its contents (18.1). This was discussed in Chapter 6 in relation to the file metaphor but it also leads to consideration of whether a more appropriate metaphor than 'save' could help the users. In this case a clear candidate might be the 'export' metaphor which is used by some other programs. For example, Adobe Photoshop 2.0 has both a 'Save as...' and an 'Export' command on its 'File' menu.

The 'export' metaphor has quite different origins from the 'save' metaphor. Whereas 'saving' implies protecting from changes, or preserving changes which have been made, 'exporting' implies sending something to another 'territory.' This distinction is exploited by Photoshop 2.0 which uses the 'Export' command when, for example, converting a file to JPEG format. JPEG compression results in the permanent loss of graphical information. Because the image is being exported to the 'JPEG world', the 'Export' command cannot be used to over-write the existing file and lose information. In Photoshop a user can make changes to an image, *save* it to a different file name with the 'Save as...' command, and then close the file. If, instead, the user makes changes to the image, *exports* it, and then attempts to close it, Photoshop does not treat the changes as having been preserved. Photoshop will not allow the user to close the file without presenting a 'Save changes before closing?' prompt because the file was exported, rather than saved.

The behaviour of Photoshop, with its separate 'Export' and 'Save as...' commands can be compared with that of Word, which attempts to use the same metaphor in all circumstances. In Word, a user can open a Word document, make changes to the formatting of the file and save it as a text file. Word then treats the changes as being saved, even though the conversion to text has abandoned all the formatting information. Indeed, because there is no separate 'Export' command, in the Macintosh version the user can even overwrite the original Word file and lose not only the format changes but all pre-existing formatting information. Consideration of the signification of the 'save' and 'file' metaphors might have helped to avoid the potential problems this raises.

7.3.4 Other types of computing metaphor

Chapter 2 introduced seven categories of metaphor which can co-exist within the design process and the interface:

Concept: Computer as theatre, interface as facade.

Design: Using metaphor as a 'tool for thought'.

Development: Work-flow, system life-cycle, object-oriented design.

Hardware: Notebook, notepad, pen, organiser.

System: Directories, menus.

Presentation: Documents, filing cabinets, rooms.

Interaction: Direct manipulation, command, conversation.

Support: Interface agents, speech bubbles.

All of these metaphors signify various concepts to the user and could thus be analysed using the LoS model. The last five categories can all be forms of interface metaphor and thus appropriate material for the model. However, it is difficult to see any immediate value in this approach for the three types at the top of the list. In these cases, the metaphors are used to generate an implementation, but will not necessarily be present in the final implementation. It is at the point of implementation that the LoS model becomes useful and it should certainly be applied to any remnants of these early metaphors which are still present. A metaphor might have been a valuable aid in the design phase, but the same metaphor could be confusing to a user whose relationship with the system is very different to that of the designer.

7.3.5 How much can the approach achieve?

Although consideration of interface metaphors prompted the work that has led to the LoS model, the development of the theory was not restricted to metaphors or to user interfaces. The LoS model is based on a semiotic approach which applies to our perception of any sign, whether it is a metaphor or not. As has been discussed above, this means that the tenor and vehicle can be independently analysed as signs in their own right and the 'What for?' method can be applied to any potential term to be used in computing, whether a metaphor or not, to examine the connotational ground.

The metaphors discussed so far have applied to specific interface elements. This is not an inherent limitation of the LoS approach. Instead, users could be questioned about the signification of a general metaphor to be used as a basis for the total interface. It is probable that useful information about, for example, the signification of desk-tops could be uncovered but this would depend on very careful phrasing of the initial question. A potential user could be asked, "What is your desk-top for?" or, "What are the objects on your desk-top for?" Obviously, the nature of the 'What for?' method is such that, if the user mentions a specific object, the questioning will continue about that object. This does not help to uncover the signification of the desk-top as a total concept for designing an interface but might help to identify the types of object, and their purposes, that could compose a desk-top interface. Consideration of the desk-top concept as a whole might be better considered by varying the initial question. This does not violate the method, in that subsequent questioning would be of the 'What for?' format. Indeed, when looking for a general metaphor, it might be better to begin with a more free-form disussion to find a concept from the user that can be used in questioning, such as 'my bit of the office'.

What the approach cannot achieve has also become clearer. The 'What for?' method is non-directed and non-exhaustive and could never form the basis of an effective design method on its own. It must be seen as an additional tool, widening the range of considerations for the analyst or designer. The greatest strength of the approach lies in the way in which it models aspects of the user's perception that are not conventionally considered; the greatest weakness of the method lies in the fact that this model remains incomplete. It could be argued that this incompleteness does not matter, in that users will always mention their most important significations and that these must therefore form the most important considerations for the designer. However, some significations will never be mentioned, an obvious example being significations that the user is unable to express verbally. However, there is currently no rival approach which offers more likelihood of uncovering these aspects – indeed no other method can even guarantee uncovering the aspects that LoS does successfully reveal. Rather than abandoning the LoS approach the answer must therefore be to continue research into the approach, seeking further validation, and to widen its scope.

7.4 Potential for further research

7.4.1 Further validation of the LoS model

The most obvious extension of this research is to interview and compare a larger number of user groups using the 'What for' technique. There would be little to be gained from larger sample sizes but comparison of more user group/interface combinations would help to isolate the effect on users' signification of different factors. This would provide further validation of the LoS model and the 'What for?' technique, in addition to information about both the interface design and user motivation. As a result of this it might be possible to refine the 'What for?' technique or to develop other tools from the LoS model.

Apart from the technique itself, the analysis of the results could certainly be improved. Rather than generic content analysis methods, analysis techniques could be developed which are more closely linked to the LoS model. These could be as simple as a check list of categories that might be expected from any interview, such as the action taken (clicking on a button, selecting a menu item), the definition of the term (as in a dictionary), the consequent action (moving on to another screen), and so forth. Such a check list could be used to immediately screen out more mundane factors, allowing anomalies to show up more easily. Other tools could be used to analyse the interview structure, identifying loops and branches in the responses, and using graphical presentation to assist the analyst's understanding of the information.

The structure of the 'What for?' interview results showed signification at various levels. Within interviews most responses moved from lower levels to higher levels. However, there is no objective measure of the signification level such that responses from two separate interviews could be ranked in relation to one another. Such a ranking would be extremely useful in comparing different implementations. A measure of this type would simplify the content analysis phase and also allow interviews to be objectively compared.

The relationship between the LoS model and GOMS was discussed above and it was suggested that the methods might be used in combination. Further research should examine ways in which the LoS model might be more tightly integrated, either with GOMS or with other equivalent low level design methods. In addition to this, the relationship of the LoS model with other high level approaches requires further study.

7.4.2 Levels of signification to the designer

It was suggested in Section 7.2.6 that, where potential users are not immediately available, designers might find some value in questioning themselves through the 'What for?' technique. However, this would only be useful if the designer sees an interface in the same way as its users will. The pilot trial of the technique, reported in Section 5.2.4, included interviews with both users and designers of interfaces which suggested that interfaces carried similar signification for both the designer and the user. The technique could be further used in this way to test whether this is always the case. In particular, it is suggested that a mis-match in signification between the designer and the user might lead to an unsuitable interface. The technique could therefore provide a valuable diagnostic for investigating systems which prove difficult to use. By interviewing both users and designers, signification mis-matches could be identified and corrected.

7.4.3 Further study of metaphors

The experiment with three metaphors reported in Chapter 4 showed that different types of metaphor can direct the user towards different levels of signification. Those results were based on open-ended questions in a questionnaire. Use of the 'What for?' technique could be used to explore this further, particularly if the results could be compared objectively as proposed above.

The LoS model also has great potential in the analysis of metaphors before their implementation. 'What for?' interviews could obviously be used to compare the signification of alternative prototype systems but, unlike many other methods, the technique can also be applied to the underlying concept. By asking users about a potential metaphor vehicle independently of its use in an interface, a 'pure' definition could be developed and used to assess the suitability of the metaphor for its context. For example, the users of the 'Navigate' command in the commercial system could have been interviewed by the designer of the system to establish what the term meant to them. Alternative metaphors which could have been used, such as 'browse' or 'go' could then be compared and the best candidate chosen. If no suitable metaphor match could be found, the structure of the system could have been changed to use, say, an index or map based alternative, before any investment in its construction had taken place.

The above discussion of the applicability of the LoS model to various categories of metaphor shows that there is no advantage to the designer in examining metaphors used in the design process. Although this might not help the designer, use of the LoS model could provide a great deal of information for researchers interested in how analysts and designers approach their work. Examining the signification of the system to the designer, particularly in terms of the metaphors used in its construction, would provide a great deal of valuable information which could be used in the development of better design methods. Indeed, once people start using the model, it could be used to analyse the 'What for?' technique and even the LoS model itself.

The studies in this thesis have focused on relatively concrete objects such as interface commands. The use of communications systems involves the user with more diffuse concepts, such as the Web. Unlike the computer systems looked at in this study, communications also involves the interaction of users, each with their own understanding of the technology. The problems of mis-matches between the signification of two users sharing the same communications facility offer a fertile area for further investigation. A particularly difficult issue here is that some of the concepts do not exist until the metaphor comes into being. The Internet existed and a number of sites on the Internet included HTML servers for some time before the concept of the 'Web' arrived; whether the concept existed before the metaphor which named it is more contentious.

The studies in this thesis have all examined particular systems at particular times, ignoring any temporal factors. As was stated in Chapter 2, metaphors gradually die, their signification therefore changing over time. Additionally, the discussion of Carroll's work drew attention to the role of metaphors in the learning process. If this is so, one would expect to see a change in the signification to a user as that user becomes more familiar with the term in its new meaning. Rather than confirming this view, the only relevant finding in my own study was that the most inexperienced user in the CUG was confused by the 'navigate' metaphor, asking whether it meant 'find'. Cornell Way's argument for metaphor as a learning aid, introduced in Chapter 2 was that, "it is easier to take parts from other established concepts than to build up new ones from scratch." (Cornell Way 1991, p.8). The example uncovered in the CUG study was perhaps an exceptional case, in that it is was likely that the term 'navigate' did not form a part of the user's normal vocabulary.

7.4.4 Wider application of the LoS model

As a semiotic approach, the LoS model is potentially applicable to any part of the communication process. Rather than examining specific metaphors or interface objects, the model could also be applied to other factors which affect the signification of the interface.

One important aspect that should be considered is the presentation of the metaphor. Presentation is particularly important to signification in being the first aspect the user is likely to notice. For example, consider the two heavily reduced screens below.

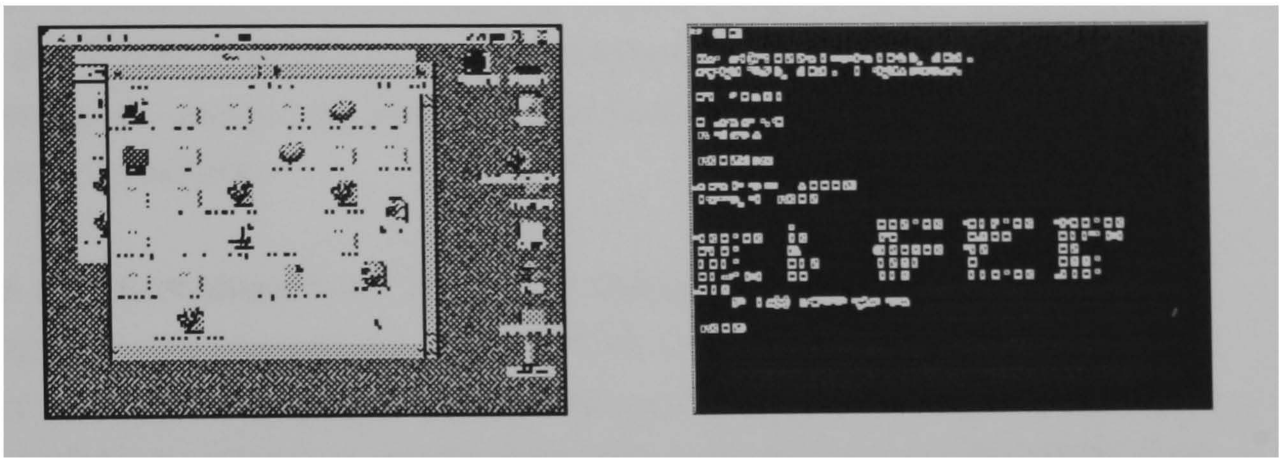


Figure 7.6: Presentation of metaphors.

It is impossible to read a word on either of these screens or to see what programs may be running but even a cursory glance immediately distinguishes the GUI screen on the left (Macintosh Finder) from the text based interface on the right (MS-DOS). Thus the overall style, or appearance, of the interface creates the initial signification to the user.

Another aspect of the presentation of the metaphor is the degree of realism. In the experiment which examined three alternative metaphors it was found that the spatial aspect of a realistically presented metaphor can be particularly powerful. Sometimes this was taken to absurd extents, such as the requests from two users for a 'remote control' for the 'television' which was always less than two inches mouse movement away. Many metaphor-based GUIs ignore these factors. Consider, for example, the comparative sizes of a Microsoft Word document and a folder on the Macintosh desktop.

Figure 7.7: Document and folder on the Macintosh desktop.

It is obvious that the Word document could not actually fit inside the folder, yet this is what the user is expected to understand. This may not matter in that the MILAN interface used a comparatively realistic perspective view of the objects, whereas the Macintosh desktop uses a flat, two-dimensional view. The LoS model could be used to examine how the levels of signification of metaphors are affected by their presentation, comparing text and graphical presentations and varying degrees of realism.

Some interface metaphors also affect the ordering of the user-system dialogue. Some researchers (Nadin 1988; Foley 1990) have suggested that direct manipulation requires a post-fix syntax, to identify the object then manipulate it, whereas text is better with a pre-fix syntax, corresponding to the typical syntax of the imperative in Indo-European languages, "Do this!" Again, the LoS model could be used to examine the effect that syntax has on the signification of the interface to the user.

7.4.5 The economic impact of the LoS model

Any factor which affects the user's performance will have an impact on the user's efficiency. The effect on performance of the factors discussed above has not been quantified. However, I carried out a simple techno-economic analysis of the potential impact of using the different categories of metaphor looked at in the experiment reported in Chapter 4. Appendix E contains an account of this analysis which was based on a techno-economic model of European industry. The model is not strong enough to be used as a reliable measure of economic impact but it does indicate that the factors considered in this thesis could have an economically significant effect on the take-up of a new technology.

References

ALLEGIANT. (1997). *SuperCard* [WWW document visited 10 July 1997] http://www.allegiant.com/supercard/

ANDERSEN, P. B. (1990). *A Theory of Computer Semiotics.* Cambridge University Press.

ANDERSON, B. (1994) et al. Minimising conceptual baggage: making choices about metaphor. *In: People and Computers IX, HCI `94.* Glasgow, August 1994, 179-194.

ANDERSSON, L-G. (1992) and Trudgill, P. *Bad language.* Harmondsworth: Penguin Books.

APPLE COMPUTER INC. (1987). *Apple human interface guidelines.* New York: Addison-Wesley Inc.

APPLE COMPUTER INC. (1989). *HyperCard stack design guidelines.* New York: Addison-Wesley Inc.

APPLE COMPUTER INC. (1991). *QuickTime design guidelines.* Cupertino, Ca: Apple Computer.

BARTHES, R. (1973b). *Mythologies.* London: Collins/ Paladin. Original publication, in French, 1957.

BARTHES, R. (1973). Myth today *In:* BARTHES, R. *Mythologies.* London: Collins/ Paladin. Original publication, in French, 1957.

BEATTIE, T. (1998). On advertising. *In: The Guardian,* 2 February 1998. *Media Guardian,* 4.

BERNSTEIN, M. (1993). Enactment in information farming. *In: ACM Hypertext'93.* New York: ACM Publications, 242-249.

BILCHEV, G. (1995) and PARMEE, I. C. The ant colony metaphor for searching continuous design spaces. *Lecture Notes in Computer Science,* No. 1993. Berlin: Springer.

BLONSKY, M. (Ed.) (1985). *On signs.* Oxford: Basil Blackwell Ltd.

BRADY, J. M. (1977). *The theory of computer science: a programming approach.* London: Chapman and Hall.

BRYMAN, A. (1988). *Quality and quantity in social research.* London: Unwin Hyman.

BUDD, R. W. (1967), THORP, R. K. and DONOHEW, L. *Content analysis of communications.* New York: Macmillan.

BURRELL, G. (1979) and MORGAN, G. *Sociological paradigms and organizational analysis.* London: Heinemann.

BUXTON, W. (1990) and MORAN, T. EuroPARC's integrated interactive intermedia facility (IIIF): early experiences. *In: IFIP WG8.4 Conference on Multi-user Interfaces and Applications.* Heraklion, Greece, 1990.

CARD, S.K. (1980a), MORAN, T.P. and NEWELL, A. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM.* 23 (7), 396-410.

CARD, S.K. (1980b), MORAN, T.P. and NEWELL, A. Computer text-editing: An information processing analysis of a routine cognitive skill. *Cognitive Psychology.* 12, 32-74.

CARR, R. M. (1991). The point of the pen. *In: Byte,* February, 1991.

CARROLL, J. M. (1982) and THOMAS, J. C. Metaphor and the cognitive representation of computing systems. *IEEE Transactions on Systems, Man and Cybernetics.* 12 (2) April 1982.

CARROLL, J. M. (1984) and MACK, R. L. Learning to use a word processor: by doing, by thinking and by knowing. *In:* THOMAS, J. C. and SCHNEIDER, M. L. (Eds.)*Human factors in computer systems.* New Jersey: Ablex.

CARROLL, J. M. (1985) and MACK, R. L. Metaphor, computing systems and active learning. *Int. Journal of Man-Machine Studies.* 22 (1) 39-57.

CARROLL, J. M. (1988), MACK, R. L. and KELLOGG, W. A. Interface metaphors and user interface design. *In:* HELANDER, M. (Ed.) *Handbook of human-computer interaction.* Amsterdam: North-Holland.

CARROLL, L. (1982). Through the looking glass. *In:* CARROLL, L. *The complete illustrated works of Lewis Carroll.* London: Chancellor Press (original private publication in 1872).

CHANDLER, D. (1995). *Semiotics for beginners.* [WWW document visited 5 November 1998]
http://www.aber.ac.uk/~dgc/semiotic.html

CHIRBAN, J. T. (1996). *Interviewing in depth: the interactive-relational approach.* London: Sage Publications.

CHOMSKY, N. (1975). *Knowledge of language: its nature, origin and use.* Westport, Connecticut: Praeger.

CHOMSKY, N. (1975). *Reflections on language.* New York: Pantheon, Random House.

COMPAQ (1987). *MS-DOS Version 3.3 reference guide.* Compaq Computer Corporation.

CONDON, C. (1990). Networked cooperative work: usability issues of MILAN (multimedia industrial local area network). *In: Telematics '90.* BIBA, Bremen, September 1990. 74-101.

CONDON, C. (1993). The computer won't let me: cooperation, conflict and the ownership of information. *In:* EASTERBROOK, E. (Ed.) *CSCW: cooperation or conflict?* London: Springer-Verlag.

CONDON, C. (1995) and KEUNEKE, S. Counting the Costs and Benefits of Metaphor. *In:* CLARKE, A. CAMPALARGO, M. KARATZAS, N. (Eds).

Bringing Telecommunications Services to the People: IS&N '95. London: Springer-Verlag.

CONDON, R. (1973). *The Manchurian candidate.* Harmondsworth: Penguin Books.

COOKE, D. J. (1984) and BEZ, H. E. *Computer mathematics.* Cambridge University Press.

CORNELL WAY, E. (1991). *Metaphor and knowledge representation.* Kluwer.

COYNE, R. (1995). Metaphors and machines. *In:* COYNE, R. *Designing information technology in the postmodern age: from method to metaphor.* Cambridge, Mass.: MIT Press, 249-301.

CRYSTAL, D. (1998). *Language play.* Harmondsworth: Penguin Books.

DE SAUSSURE, F. (1974). *Course in general linguistics.* London: Fontana/Collins (originally published in French, 1915).

DE SOUZA, C. S. (1993). The semiotic engineering of user interface languages. *International Journal of Man-Machine Studies.* 39, 753-773.

DIEBERGER, A. (1994a). *Navigation in textual virtual environments using a city metaphor.* Ph.D. Thesis, Vienna University of Technology

DIEBERGER, A. (1994b) and TROMP, J. G. The information city - a metaphor for navigating hypertexts. *In: People and Computers VIII: HCI `94.*, Loughborough, 1993, 179-194.

DOUGLAS, S. A. (1983) and MORAN, T. P. Learning text editor semantics by analogy. *In: CHI '83: Human Factors in Computing Systems.* New York: ACM Publications.

DUIN, H. (1994) and KIESSLING, S. *CRIMP for Windows 3.1 User Manual.* Bremen, Germany: BIBA.

DUIN, H. (1995) Object-Oriented Scenario Management for Simulation Models. *In: IMACS European Simulation Meeting on Simulation Tools and Applications*. 28-30 August, 1995, Gyor, Hungary.

ECO, U. (1979). *A theory of semiotics*. Midland Book Edition. Bloomington, Indiana: Indiana University Press.

ECO, U. (1985). The semantics of metaphor. *In:* INNIS, R.E. (Ed.) *Semiotics: an introductory reader*. Bloomington, Indiana: Indiana University Press.

ECO, U. (1987). *Travels in hyperreality*. London: Pan Books.

EISENBERG, A. (1992). Metaphor in the language of science. *In: Scientific American*. May 1992, 144.

EUROSTAT (1993) *Enterprises in Europe: Second Report*. Brussels: Commission of the European Communities.

FEYNMAN, R. P. (1990). *QED: The strange theory of light and matter*. Harmondsworth: Penguin Books.

FISCHER, G. (1991). The importance of models in making complex systems comprehensible. *In:* TAUBER, M.J. and ACKERMANN, D. (Eds.) *Mental models and human computer interaction* 2. Amsterdam: Elsevier Science Publishers, BV, North-Holland.

FLEXNER, S. B. (Editor-in chief.) (1987).*The Random House dictionary of the English language*. 2nd Edition. New York: Random House Dictionaries.

FLORIN, F. (1990). Information landscapes. *In:* AMBRON, S. and HOOPER, K. (Eds.) *Learning with interactive multimedia*. Redmond, Washington: Microsoft Press, 27-49.

FOLEY, J. D, (1990), VAN DAM, A. et al. *Computer graphics: principles and practice*. 2nd Edition. New York: Addison-Wesley Inc.

FREKSA, C. (1991). Qualitative spatial reasoning. *In:* MARK, D. M. and FRANK, A. U. (Eds.) *Cognitive and linguistic aspects of geographic space.* Kluwer, 361-372.

GARDNER, M. (1992). Charles Sanders Peirce. *In:* GARDNER, M. *Fractal music, Hypercards and more.* New York: W. H. Freeman.

GAVER, W. (1992) et al. Realizing a video environment: EuroPARC's RAVE system. *In: Proceedings of CHI'92.* New York: ACM publications, 27-35.

GEOCITIES. (1998). *GeoCities* [WWW document visited 5 November 1998] http://www.geocities.com/

GERRIG, R. (1983) and HEALY, A. Dual processes in metaphor understanding: comprehension and appreciation. *Journal of Experimental Psychology: Learning, Memory and Cognition.* 9, 667-675.

GILL, J. (1997) and JOHNSON, P. *Research methods for managers.* 2nd Edition. London: Paul Chapman.

GOATLY, A. (1997). *The language of metaphors.* London: Routledge.

GORDEN, R. L. (1987). *Interviewing : strategy, techniques, and tactics.* 4th Edition. Chicago : Dorsey Press.

GORDON, T. J. (1968) and HAYWARD, H. Initial Experiments with the Cross Impact Matrix Method of Forecasting. *In: Futures.* December 1968, 100-116

GRUNDY, F. (1996). *Women and computers.* Exeter: Intellect Books.

HALASZ, F. (1982) and MORAN, T.P. Analogy Considered Harmful. *In: Conference on Human Factors in Computer Systems.* Gaithersburg, Maryland, USA, 1982.

HALL, S. (1980). Encoding/decoding. *In:* CENTRE FOR CONTEMPORARY CULTURAL STUDIES (Ed.) *Culture, media, language.* London: Hutchinson.

HÄMMÄINEN, H. (1991) and CONDON, C. Form and room: metaphors for groupware. *In: COCS '91, ACM Conference on Organizational Computer Systems.* Atlanta, Georgia, 5-8 Nov 1991.

HAMMOND, N. (1987) and ALLINSON, L. The travel metaphor as design principle and training aid for navigating around complex systems. *In:* DIAPER, D. and WINDER, R. (Eds.) *People and computers III: proceedings of HCI'87.* 75-90.

HELMER, O. Problems in Futures Research: Delphi and Causal Cross-Impact Analysis. *In: Futures.* February 1977, 17-31.

HENDERSON JR, D. A. (1986). and CARD, S. K. Rooms: the use of multiple virtual workspaces to reduce contention in a window-based graphical user interface. *In: ACM Transactions on Graphics.* 5 (3) July 1986, 211-243.

HERSEN, M. (1976). and BARLOW, D. H. *Single case experimental designs: strategies for studying behaviour change.* Oxford: Pergammon.

HESSE, M. (1980). The explanatory function of metaphor. *In*: HESSE, M. *Red the philosophy of science.* Brighton, Sussex: Harvester, 111-24.

HESTER, M. B. (1967). *The meaning of poetic metaphor.* Mouton, The Hague.

HOFFMAN, R. R. (1984). Recent psycholinguistic research on figurative language. *In:* WHITE, S. (Ed.) *Discourses in reading and linguistics, annals of the NY Academy of Sciences.* 433, 137-166.

HOOPER, K. (1986). Architectural design: an analogy. *In:* NORMAN, D. A. and DRAPER, S. W. (Eds.) *User centred system design: new perspectives on human-computer interaction.* Hillsdale, N-J: Lawrence Erlbaum Associates.

HUCZYNSKI, A. (1991). and BUCHANAN, D. *Organizational behaviour: an introductory test.* 2nd Edition. Hemel Hempstead: Prentice Hall International.

HUTCHINS, E. (1989). Metaphors for interface design. *In:* TAYLOR, M.M. et al (Eds.) *The structure of multimodal dialogues.* Amsterdam: Elsevier, North Holland.and

HUTCHINSON, C. (1993). and ROSENBERG, D. Cooperation and conflict in knowledge-intensive computer supported cooperative work. *In:* EASTERBROOK, E. (Ed.) *CSCW: cooperation or conflict?* London: Springer-Verlag.

ISBISTER, K. (1995) and LAYTON, T. Agents: What (or who) are they? *In:* NIELSEN, J. (Ed.) *Advances in Human-Computer Interaction, vol. 5.* Norwood, N.J.: Ablex.

JACKSON, P.J. (1995). *Information systems as metaphor: innovation and the 3Rs of representation.* Internal paper. Department of Management Studies, Brunel University.

JAKOBSON, R. (1956). Two aspects of language and two types of linguistic disturbances. *In:* JAKOBSON, R. and HALLE, M. (Eds.) *Fundamentals of Language.* The Hague.

JOHN, B. E. (1966) & KIERAS, D. E. The GOMS family of user interface analysis techniques: comparison and contrast. *ACM Transactions on Computer-Human Interaction.* 3(4) 320-351.

JOHNSON, J. (1987). How faithfully should the electronic office simulate the real one? *SigCHI Bulletin.* 19 (2) October 1987 21- 24.

JOHNSON-LAIRD, P. N. (1983). *Mental models.* Cambridge University Press.

KANE, J. A Primer for a New Cross-Impact Language - KSIM. *In: Technological Forecasting and Social Changes.* 4, 1972, 129-142.

KAWASAKI, Guy. (1998). *Evangelist.* [WWW document visited 5 November 1998] http://www.evangelist.macaddict.com/

KAY, A. (1984). Computer software. *In: Scientific American,* Sept. 1984, 251 (3) 52-59.

KAY, A. (1990). User interface: a personal view. *In:* Laurel, B. and Mountford, S. J. (Eds.) *The art of human computer interface design.* New York: Addison-Wesley Inc.

KELLY, M. H. (1987) and KEIL, F. C. Metaphor comprehension and knowledge of semantic domains. *Journal of Metaphor and Symbolic Activity.* 2 (1) 33-51.

KENTUCKY (1998) *University of Kentucky Classics Department: Rhetorical Figures.* [WWW document, visited 23/10/98] http://cal.bemidji.msus.edu/English/Resources/RhetFigures.html

KILGANNON, P. (1980). *Business data processing and systems analysis.* London: Edward Arnold.

KRAUTH, J. (1994), HAMACHER, B. and DUIN, H. A Set of Tools for Rough Enterprise Modelling. *In: Proc CISS (First Joint Conference of Simulation Societies).*

KRIPPENDORF, K. (1980). *Content analysis: an introduction to its methodology.* London: Sage Publications.

KUHN, W. (1991) and FRANK, A. U. A formalization of metaphors and image-schemas in user interfaces. *In:* MARK, D. M. and FRANK, A. U. (Eds.) *Cognitive and linguistic aspects of geographic space.* Kluwer, 419-434.

LAING, R. D. (1966), PHILLIPSON, H. and LEE A. R. *Interpersonal perception.* London: Tavistock Publications.

LAING, R. D. (1970). *Knots.* London: Tavistock Publications.

LAKOFF, G. (1980). JOHNSON, M. *Metaphors we live by.* University of Chicago Press.

LAKOFF, G. (1987). *Women, fire and dangerous things: what categories reveal about the mind.* University of Chicago Press.

LAUREL, B. (1986). Computers as mimesis. *In:* NORMAN, D. A. and DRAPER, S. W. (Eds.) *User centred system design: new perspectives on human-computer interaction.* Hillsdale, N-J: Lawrence Erlbaum Associates.

LAUREL, B. (1990). Interface agents: metaphors with character. *In:* Laurel, B. and Mountford, S. J. (Eds.) *The art of human computer interface design.* New York: Addison-Wesley Inc.

LAUREL, B. (1993). *Computers as theatre.* New York: Addison-Wesley Inc.

LEATHERDALE, W. H. (1974). *The role of analogy, model and metaphor in science.* Amsterdam: North-Holland.

LÉVI-STRAUSS, C. (1969). *Totemism.* Harmondsworth: Penguin (original publication in French, 1962).

LEVIDOV, L. (1989) and ROBINS, K. (Eds.) *Cyborg worlds: the military information society.* London: Free Association Books.

LEVIN, S. R. (1988). *Metaphoric worlds: conceptions of a romantic nature.* New Haven, CT: Yale University Press.

LIEBERMAN, H. (1997). Autonomous Interface Agents. *In: Proceedings of CHI '97.*. New York: ACM Publications.

LODGE, D. (1977). *The modes of modern writing: metaphor, metonymy, and the typology of modern literature.* London: Edward Arnold.

LODGE, D. (1990). Narration with words. *In:* BARLOW, H. et al (Eds.) *Images and understanding.* Cambridge University Press.

LUMBRERAS, M. F. (1993). A hypertext for blind people. (Poster) *In: Hypertext'93.* Seattle, 1993

MARCUS, A. (1983). Graphic design for computer graphics. *IEEE computer graphics and applications.* 3(4), July 1983, 63-70.

MARTIN, G. D. (1975). *Language, Truth and Poetry.* Edinburgh University Press.

MEEHL, P. E. (1978). Theoretical risks and tabular asterisks: Sir Karl, Sir Ronald, and the slow progress of soft psychology. *Journal of Consulting and Clinical Psychology.* 46, 806-35.

MITS (1993). *Deliverable D4: Draft Metaphor Specifications.* RACE Project R2094, DG XIII, European Commission, Brussels.

MITS (1994). *Deliverable D7: Report on Demonstrators.* RACE Project R2094, DG XIII, European Commission, Brussels.

MYNATT, E. D. (1994) and WEBER, G. Nonvisual presentation of graphical user interfaces: contrasting two approaches. *In: CHI '94.* ACM Publications, 166-172.

NADIN, M. (1988). Interface design and evaluation – semiotic implications. In: HARTSON, H. R. and HIX. H. (Eds.) *Advances in human-computer interaction vol.* 2. New Jersey: Ablex

NCC (1978). *Introducing systems analysis and design: Volume 1.* Manchester: NCC Publications.

NELSON, T. H. (1990). The right way to think about software design. *In:* Laurel, B. and Mountford, S. J. (Eds.) *The art of human computer interface design.* New York: Addison-Wesley Inc.

NETSCAPE (1988) *Netscape Navigator 4.0.* [WWW document visited 5 November 1998] http://www.netscape.com/

NIETZSCHE, F. (1974). *Early Greek philosophy and other essays. The complete works of Friedrich Nietzsche,* Vol 2. Ed. LEVY, O. New York: Gordon Press.

NORMAN, D. A. (1986a). Cognitive engineering. *In:* NORMAN (1986b).

NORMAN, D. A. (1986b) and DRAPER, S. W. (Eds.) *User centred system design: new perspectives on human-computer interaction.* Hillsdale, N-J: Lawrence Erlbaum Associates.

OGDEN, C. K. (1938) and RICHARDS, I. A. *The meaning of meaning: a study of the influence of language upon thought and of the science of solipsism.* New York: Harcourt Brace.

OREN, T. (1990) et al. Guides: characterizing the interface. *In:* Laurel, B. and Mountford, S. J. (Eds.) *The art of human computer interface design.* New York: Addison-Wesley Inc.

ORTONY, A.(1978) et al. Metaphor: theoretical and empirical research. *Psychological Bulletin.* 85 (5), 919-943.

PEIRCE, C. S. (1985). Logic as semiotic: the theory of signs. *In:* INNIS, R. E. (Ed.) *Semiotics: an introductory reader.* Bloomington, Indiana: Indiana University Press.

PEJTERSEN, A. M. (1988) and GOODSTEIN L. P. Beyond the Desk Top metaphor: Information Retrieval with an Icon-Based Interface. *In: Workshop Visualization in Human computer interaction, 7th interdisciplinary workshop informatics & psychology,* Schärding 1988. 149-182.

PEMBERTON, L. (1993). Offices, balconies, doors and corridors: an experimental interface metaphor for integrating collaborative design styles. *Lecture notes in computer science: Human Computer Interaction: Vienna Conference, VCHCI '93.* Vol. 733, 419-420.

PIAGET, J. (1971). *Structuralism.* London: Routledge and Kegan Paul.

PIMENTA, M. S. (1997) and FAUST, R. HCI and Requirements Engineering - Eliciting Interactive Systems Requirements in a Language-Centred User-Designer Collaboration: A Semiotic Approach. *SIGCHI Bulletin.* 29(1).

PRATES, R. O. (1997), DE SOUZA, C. S. and GARCIA, A. C. B. G. A semiotic framework for multi-user interfaces. *SIGCHI Bulletin.* 29(2), 28-39.

QUINTANAR, L. R. (1982), CROWELL, C. R. and PRYOR, J. B. Human-computer interaction: a preliminary social psychological analysis. *Behaviour Research Methods & Instrumentation.* 14 (2), 210-220.

RAUCH, T. (1997) et al. Enabling the book metaphor for the World Wide Web: disseminating on-line information as dynamic Web documents. *IEEE transactions on professional communication.* 40 (2), 111.

RICH, C. (1996) Window Sharing with Collaborative Interface Agents. *SIGCHI Bulletin.* 28 (1), 70-78.

RICHARDS I. A. (1936). *The philosophy of Rhetoric.* Oxford University Press.

RIFAS L. (1994). *The dataforest - tree forms as information display graphics.* Dissertation, University of Washington.

ROBSON, C. (1973). *Experiment, design and statistics in psychology.* Harmondsworth: Penguin Books.

ROBSON, C. (1993). *Real world research: a resource for social scientists and practitioner-researchers.* Oxford: Blackwell.

ROGERS, Y. (1988), LEISER, R. and CARR, D.,. Evaluating metaphors for use at the user-system interface. *In:* Bullinger, H-J. et al. (Eds.) *Eurinfo '88, proceedings of the First European Conference on Information Technology For Organisational Systems.* Athens, Greece, 16-20 May, 1988. Elsevier Science Publishers B.V, North-Holland.

ROOT R. W. (1988). Design of a multi-media vehicle for social browsing. *In: SCW 88.* Portland, Oregon, Sept. 26-28 1988, 25-38.

RUBIN, J. (1995) and RUBIN, I. S. *Qualitative interviewing : the art of hearing data.* Thousand Oaks, California: Sage Publications

SACKS O. (1985). *The man who mistook his wife for a hat.* London: Duckworth.

SACKS, O. (1989). *Seeing voices.* University of California Press.

SCHWARZ, C. (1988) et al. *Chambers English Dictionary*. Edinburgh: W & R Chambers Ltd.

SEBEOK, T. A. (1991). *A sign is just a sign*. Bloomington: Indiana University Press.

SHNEIDERMAN, B. (1987). *Designing the user interface: strategies for effective human-computer interaction*. Reading, Mass: Addison-Wesley.

SIDMAN, M. (1960). *The tactics of social research*. New York: Basic Books.

SIEGEL, S. (1988) and GALLAGHER, N. J., Jr . *Nonparametric statistics for the behavioural sciences*. 2nd Edition. New York: McGraw-Hill.

SINNIGEN, M. (1994), COOLS, J-P. and SAMPERA, J. *URSA Deliverable 5*. RACE Project R2091, DG XIII, Brussels.

SMITH, D. C. (1982) et al. Designing the Star user interface. *In: Byte*. 7(4), April 1982, 242-282.

SMYTH, M. (1995a), ANDERSON, B. and ALTY, J.L. Metaphor reflections and a tool for thought. *In: People and Computers X. HCI '95*. Huddersfield, Cambridge University Press, 137-150.

SMYTH, M. (1995b), ANDERSON, B., KNOTT, R. and ALTY, J.L. Reflections on the design of interface metaphors. *In: Interact '95*. Lillehammer, 1995.

SMYTH, M. (1993) and KNOTT, R. The role of metaphors at the human computer interface. *In: VAMMS '93, Visual Aspects in Man-Machine Systems*. Prague, Czech Republic, 24-27 March, 1993. Ellis-Horwood, In Press.

Spiliotopoulos, V. (1981) and Schackel, B. Towards a computer interview acceptable to the naive user. *International Journal of Man-Machine Studies*. 14, 77-90.

STURROCK, J. (1986). *Structuralism*. London: Paladin, Collins.

TAUBER, M.J. (1991) and ACKERMANN, D. (Eds.) *Mental models and human computer interaction* 2. Amsterdam: Elsevier Science Publishers, BV, North-Holland.

THIMBLEBY, H. (1991). Can anyone use the video? *In: New Scientist.* 129 (1757) (23/02/91), 48 - 51.

VICTORIA (1998) *The University of Victoria, Canad, Writer's Guide: Literary and Rhetorical Terms By Category.* [WWW document, visited 27/10/98] http://elza.lpi.ac.ru/WritersGuide/mastertoc.html [original WWW document, visited 7/8/96], now defunct, http://webserver.maclab.comp.uvic.ca/writersguide/Literary/LiteraryTermsTOC.html

VIHMA, S. (Ed.) (1990). *Semantic visions of design: proceedings from the Symposium on Design Research and Semiotics, 17-18.5.89.* Helsinki: University of Industrial Arts.

VIRTUS (1997). *Walkthrough.* [WWW document visited 5 November 1998] http://www.virtus.com/

WEINREICH, U. (1966). On the semantic structure of language. *In:* GREENBURG, J. H. *Universals of language.* 2nd Edition. MIT Press, 142-216.

WELLEK, R. (1976) and WARREN, A. *Theory of literature.* Harmondsworth: Penguin Books.

WHITTOCK, T. (1990). *Metaphor and film.* Cambridge University Press

WIENER, N. (1961). *Cybernetics.* 2nd Edition. MIT Press.

YEATES, D. (1991) (Ed.) *Project management for information systems.* London: Pitman Publishing.

YOUNG, R. M. (1983). Surrogates and mappings: two kinds of conceptual models for interactive devices. *In:* GENTNER, D. and STEVENS, A. L. *Mental Models.* London: Lawrence-Erlbaum.

ZEISEL, J. (1984). *Inquiry by design.* Cambridge University Press

Appendix A: Tropes

A.1 Introduction

There are two types of rhetorical device or figure: schemes and tropes. Schemes use words in their literal sense but alter the grammar and/or vocabulary to achieve dramatic effect. An example is alliteration, where words are selected for similarity of sound such as starting with 's', as in the previous phrase. Tropes are devices which use words in other than their literal sense, such as metaphor and metonymy, which have been considered in this thesis. The following list places these in context by listing other tropes my research uncovered, together with some examples of their use. The distinction between a trope and a scheme is not always clear and some rhetorical figures have been included which are not truly tropes but might be used in a similar manner. The list has been composed and edited from a number of sources, mainly on the World Wide Web, principally the lists produced by the Universities of Kentucky (1998) and Victoria (1998), together with Chambers 20th Century Dictionary (Schwarz 1988).

A.2 Some rhetorical devices

Apostrophe

A sudden turn from the general audience to address a specific group or person or personified abstraction absent or present: For Brutus, as you know, was Caesar's angel (addressed to the other characters). Judge, O you gods, how dearly Caesar loved him (addressed to the gods). (Shakespeare, Julius Caesar).

Archaism

Use of an older or obsolete form. Not a true trope, but can be used in a non-literal sense, as when a pub is given an 'Olde Worlde' name to imply a long history.

Catachresis

A harsh metaphor involving the use of a word beyond its strict sphere: I listen vainly, but with thirsty ear. (MacArthur, Farewell Address).

Euphemism

Substitution of an agreeable or at least non-offensive expression for one whose plainer meaning might be harsh or unpleasant. Not necessarily a trope but a device which often involves tropes such as metaphor.

Hyperbole

Exaggeration for emphasis or for rhetorical effect: a million examples come to mind in a second.

Irony

The term irony is derived from the Greek eiron (dissembler), and denotes that the appearance of things differs from their reality, whether in terms of meaning, situation, or action. It usually involves the expression of something which is contrary to the intended meaning; the words say one thing but mean another: Yet Brutus says he was ambitious; And Brutus is an honourable man. (Shakespeare, Julius Caesar).

Litotes

Understatement, especially by denying the contrary of the thing being affirmed: Richard Branson's wealth is not insignificant. (Sometimes used synonymously with meiosis, sometimes treated as a special case of meiosis.)

Meiosis (understatement)

Purposefully representing a thing as much less significant than it is, achieving an ironic effect: A nuclear bomb can ruin your whole day.

Metaphor and metonymy

These have been defined in chapter 2.

Oxymoron

Apparent paradox achieved by the juxtaposition of words which seem to contradict one another: I must be cruel only to be kind. (Shakespeare, Hamlet).

Paradox

Not truly a trope, but an assertion seemingly opposed to common sense that may yet have some truth in it when taken in a metaphorical sense: What a pity that youth must be wasted on the young. (George Bernard Shaw).

Personification (Prosopopeia)

Personification is the attribution of human (or occasionally animal) qualities to inanimate objects or abstract concepts. When we speak of jealousy "rearing its ugly head," we are personifying jealousy by giving it animate form.

Rhetorical Question

A rhetorical question implies that the answer is obvious – the kind of question that does not need actually to be answered. It can be used for rhetorically persuading someone of a truth without argument: "Is the Pope a Catholic?" It can also be used as a trope to give emphasis to a supposed truth by stating its opposite ironically: "You are joking, aren't you?"

Simile

Differs from metaphor in making the comparison between two things explicit by using 'like' or 'as'.

Synecdoche

Understanding one thing with another; the use of a part for the whole, or the whole for the part. (A form of metonymy.)

Appendix B: Semiotic analysis

B.1 Introduction

As part of this exercise, I deliberately kept trying to look at the signification from a range of viewpoints, including a user, an external observer, Apple itself and the organisation or institution in which the user is working. I also allowed my mind to wander, considering corollary significations. It is not suggested that this is typical of the significations an interface designer might uncover, nor that this would concord with an analysis carried out by an expert semiotician, only that this is what is possible for a someone such as myself who has never previously carried out a semiotic analysis.

B.2 Signification uncovered

The following significations are numbered. The numbering refers to the order in which they were generated. Sometimes a signification was seen to have two aspects, labelled 'a' and 'b'. Although the significations tend to move from what appear to be the lowest to the highest levels, this can only be a subjective judgment and is not consistent. Some significations were generated by considering the signification from a changed viewpoint which might generate another signification at the same or a lower level.

As an external observer:

1 The words carry a simple meaning as part of a sentence within a textual context not visible in this fragment.

2 'This is a sentence in the English language'

3 'This is being written by someone who has received a reasonable education in this language'

Note that the truth of the statement is not relevant to this form of analysis. The user might have laboriously constructed a sentence in an unfamiliar language in order to falsely signify this meaning. This does not make it a non-signification; Eco has pointed out that the ability to lie is at the very heart of semiotics.

> '*Every time there is a possibility of lying, there is a sign-function* : which is to signify (and then to communicate) something to which no real state of things corresponds. A theory of codes must study everything that can be used in order to lie. The possibility of lying is the *proprium* of semiosis...'
> (Eco 76, pp 58-59, italics in original)[1]

4 'The person currently manipulating this Macintosh is (has just been/ will be) changing this message'

From the user's point of view:

5 'This person is not satisfied with the statement as it stands',

6 'This user has some understanding of how to use a word processor'

However, the user could be deleting the wrong word:

7 'The meaning of this statement does not matter to the Macintosh'

As an external observer:

8 'This is a word processor'

---

1 Eco, U. 'A Theory of Semiotics', Indiana University Press, Bloomington, 1976

---

9 'This is this designer's interpretation of how the user should perceive this utility'

10 'This is a particular style of graphical user interface'

11 'This environment was expensive to produce'

12a 'It is worth the extra expense', or **12b** 'It is worthwhile'

From Apple's point of view:

13 'Apple care about the ease of use of your computer',

14 'Apple care about <u>you</u> !'

Or from an external observer/ critic:

15 'We at Apple do not think that you are as clever as we are, so we have made an interface which can be used by an idiot'

16 'We think that you, the user, are an idiot'

From the user's view in choosing to use a Mac:

17a 'I am a novice', or **17b** 'I am a maverick; I don't care what you think of me'

18 'I am willing to go out on a limb',

Which leads to consideration of an IBM PC user:

19a 'Nobody ever got sacked for buying IBM' or **19b** 'Nobody ever got promoted for buying IBM'

For an organisation using Macs:

20 'We encourage originality (or peculiarity)',

or, one using IBMs:

21a 'We don't like people to make rash decisions', or **21b** 'We stifle originality'

22a 'We are a really go-ahead company', or **22b** 'We don't actually know where we are going'

Equally, the total absence of computers in a modern office environment becomes a statement:

23 'We want to carry on doing things the way we always have'

Where a company is conscious of what it is saying at this level, this again moves us up to the next level:

24 'Here is our corporate image'

25 'You are dealing with a large corporation which means business

From the external observer looking at the organisation, this could become:

26a 'A triumph for Western capitalism', or **26b** 'A symbol of Western decadence'

At this point I ran out of ideas about where to go next.

Appendix C: Experiment results

C.1 Questionnaire responses

For each subject, tables were drawn up containing the answers to each question. These were converted from Yes/No answers into correct and incorrect responses and were then sorted into the four categories of question. Within each of these categories, the responses by a given subject were combined. The mean confidence rating for each category, the standard deviation and the percentage of correct answers were then calculated, giving the results shown in the three tables below:

Table C.1: MILAN results.

	Confidence Ratings		Correct Answers	
Category	**Mean**	**S.D.**	**Total**	**% age**
$S \cap V$	89.4	22.6	14	42
$S \cap \sim V$	73.4	27.3	22	66
$\sim S \cap V$	61.7	34.2	19	57
$\sim(S \cup V)$	74.2	23.8	8	24
Overall	74.7		63	47

Table C.2: Link Journal results.

	Confidence Ratings		Correct Answers	
Category	Mean	S.D.	Total	% age
S∩V	59.8	30.5	19	57
S∩~V	55.6	31.0	18	54
~S∩V	45.5	34.5	12	36
~(S∪V)	62.4	31.9	14	42
Overall	55.8		63	47

Table C.3: Little People results.

	Confidence Ratings		Correct Answers	
Category	Mean	S.D.	Total	% age
S∩V	72.7	27.5	23	69
S∩~V	62.9	30.1	26	78
~S∩V	53.0	28.5	14	42
~(S∪V)	44.7	33.5	13	39
Overall	58.3		76	57

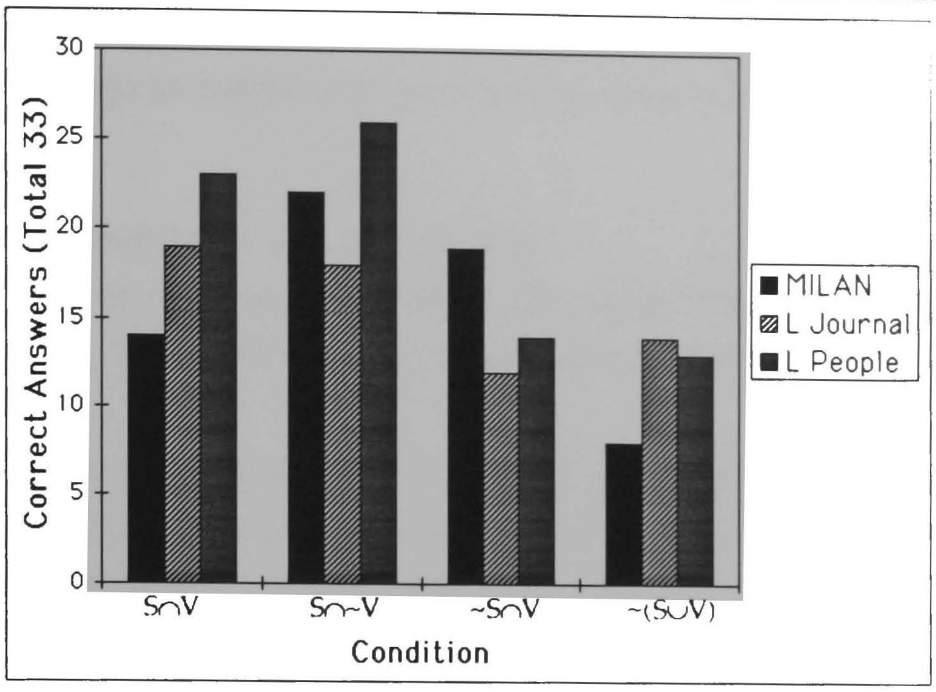


Figure C.1: Summary of correct responses.

C.2 Analysis of results

In the experiment by Anderson et al., the number of correct responses for the ~S∩V case was significantly lower than any of the other three cases for all the interfaces, leading to their conclusion that conceptual baggage was the main source of errors. As can be seen from the graph above, this does not accord with the findings of my experiment, in which the ~(S∪V) case led to most errors overall. Also, unlike the previous experiment, the standard deviation between the three systems for the ~S∩V case was not higher than the others, as can be seen from the standard deviation given below. Although the standard deviation is only based on three systems in each condition and the actual values should not be accorded too much importance, there is certainly no evidence of results for the ~S∩V case deviating more than the others.

Table C.4: Standard deviation for each category.

Condition	S∩V	S∩~V	~S∩V	~(S∪V)
S.D.	3.68179	3.266	2.9439	2.6247

Finally, as it was possible that the subjects had attached different confidence ratings to correct and incorrect answers, a weighted average was generated:

Weighted average = $(\Sigma(CR*CA))/N$

where CR is confidence rating, in the range 0-100,
CA is correct answer (value 1 for correct, -1 for incorrect)

The resulting value, on a scale from -100 to 100 was then transposed to a 0 to 100 scale, giving values equivalent to those in the unweighted results. The resulting weighted values were:-

MILAN	48.58
Link Journal	51.59
Little People	48.58

As can be seen, the only result of this exercise was to bring the figures even closer to the value of 50 one would expect from random answers. The following table shows the results for the four categories:

Table C.5: Weighted results for each category.

	Confidence Ratings		Correct Answers	
Category	**Mean**		**Total**	**% age**
S∩V	74.0		54	56
S∩~V	64.0		66	66
~S∩V	53.4		45	45
~(S∪V)	60.4		35	35
Overall	63.0		200	50

Appendix D: Content analysis

D.1 Introduction

D.1.1 Presentation of the data

The first column lists the response elements gathered into the categories allocated by the two evaluators. The second column shows the category tags allocated to the elements in the first stage of analysis, whilst the third shows the tags of all other statements allocated to the same category in the second stage of the content analysis. The same abbreviations are used in this appendix as in chapter 6:

x.y	Category references, where x is the category tag allocated by the evaluator and y is the number of the interviewee.
CUx	Commercial User x (where x=1 to 5).
AUx	Academic User x (where x=1 to 5).
MSE	Media Studies Evaluator
CSE	Computing Studies Evaluator

Small changes have been made to preserve the anonymity of the commercial organisation: references to the company name have been replaced by 'COMPANY' and the name of the computer system has been replaced by 'SYSTEM'.

D.1.2 Organisation of the data

The results of the content analysis are shown in the third column of the results tables. Each evaluator compared the tagged response elements from the second interview within a user group with the elements from the first interview, marking elements which belonged to the same category; the third interview was then compared with the combined categories from the first and second interviews, and so forth.

In theory, it might be supposed that the content analysis stage would consist of the assignment of a category identified in one set of responses as equivalent to a category identified in an earlier set of responses. However, the actual results were are not so straightforward, in that sometimes elements tagged as part of the same category in the first stage were then allocated to different categories in the content analysis stage. Consider, for example, part of the content analysis by CSE for AU2:

Response element	**Tag**	**Category**
...or if I want to save it...	6.2	1.1
...as another format...	6.2	7.1
...that I can use...	7.2	-
...with another tool.	7.2	13.1

Originally, the evaluator identified two significations for these response elements (all parts of a single response from the interviewee):

6.2 "...or if I want to save it as another format..."

7.2 "...that I can use with another tool.

When comparing user AU2 with AU1, however, the evaluator re-adjusted the categories. For category 6.2, the concept of saving – "or if I want to save it" – was separated from the format changing aspect – "as another format". These were respectively marked as belonging to two separate categories from AU1: category 1.1 – "It's for saving" – and category 7.1 – "to another format". This might appear to be inconsistent but is not necessarily so. Originally the phrase had been tagged as a single concept – "or if I want to save it as another format", but the evaluator had already identified the saving signification as category 2.1 – "That's for saving a file". The new concept introduced was that of changing format and the entire phrase was placed in that category. When examining individual elements against the categories from AU1, however, both "or if I want to save it" and "That's for saving a file" were allocated to category 1.1. This does not affect the overall content analysis results in that the identification of common categories between the two interviewees is the same as if both parts of 6.2 had been tagged as belonging to category 7.1.

Although appearing to be a simpler case, category 7.2 does represent a problem. Only the second element – "with another tool." – is marked as belonging to one of the AU1 categories. This leaves the first part – "that I can use" as a signification on its own. The question arises as to whether this is truly a separate signification. CSE originally allocated it the same tag as "with another tool" as part of the same phrase and its meaning certainly supports this interpretation. Other examples of this practice are more extreme: for example, in CSE's analysis of AU3's responses, the phrase "Yes, to save a file" has been treated as a single signification, tagged 2.3, but only the second part has been identified as belonging to category 1.1, leaving the phrase "Yes" as a signification on its own.

The converse has also taken place, in that elements originally distinguished as separate in the first stage of analysis have been combined in the second stage. Consider, for example, the following extract from CSE's analysis of AU3's responses:

Response element	**Tag**	**Category**
And because I cannot remember everything.	10.3	
I don't have to re-type everything.	11.3	20.1
That is one of the main problems,...	12.3	
...points,...	12.3	
...why I would actually be saving it.	12.3	
I don't want to repeat the same effort...	13.3	20.1
...again,...	0.3	

In this case, the phrases "I don't have to re-type everything" and "I don't want to repeat the same effort " were originally given separate tags, 11.3 and 13.3 respectively. However, when comparing them to the combined categories from AU1 and AU2, evaluator CSE has identified them both as belonging to category 20.1, originally based on AU1's response "So that you might not repeat yourself". The most obvious explanation is that the evaluator simply failed to spot the equivalence in the category formation stage but did identify it in the content analysis stage.

The results shown in the appendix have been adjusted to take account of these anomalies. The combination is purely objective, based on two simple rules:

> All cases in which an evaluator has tagged elements as belonging to the same category will stand, unless they have been explicitly allocated to separate categories in the content analysis stage.
>
> All cases in which the evaluator has tagged elements from different categories as belonging to the same category in the content analysis stage will be treated as combining those categories.

Elements tagged '0' as having no signification have been removed and common categories have been fully cross-referenced. For ease of comparison, the same numbering system has been used. Where a category has been split under the first rule, the second category is identified by the suffix 'a' (no category needed to be split more than once).

D.2 MSE content analysis

D.2.1 Commercial User Group

Table D.1: Content analysis of CU1

Response element	**Cat-egory**	**Common categories**
When you go... You go into 'Navigate'...	1.1	3.2,2.3,5.4,2.5
...into that particular screen.	2.1	6.4
...to input an invoice. To input invoices. We input invoices...	3.1	9.4,5.5
That's the first button you press,...	4.1	2.2
...before going on to the next fields.	5.1	
...to bill...	6.1	6.3
...one part of COMPANY... ...to another part of COMPANY. Because if one part of COMPANY... ...is doing work... ...or providing services... ...to another part of COMPANY... ...then they need to be charged for it. ...that each part of COMPANY...	7.1	12.4
So that the books at the end of the day are...	8.1	
I won't say one hundred percent correct - ...but as correct as they possibly can be...	9.1	
...for each part of the business.	10.1	
It's a requirement...	11.1	

...under legislation and under Oftel...	12.1	
...should be responsible...	13.1	
...for their own accounting purposes.	14.1	10.5

Table D.2: Content analysis of CU2

Response element	Category	Common categories
...you can do anything with that.	1.2	
Just press... ...in the first place.	2.2	4.1
...the 'navigate'... So you can navigate... ...around the whole system. So it'll get you... ...without using the 'navigate' button...	3.2	1.1,2.3,5.4,2.5
...and then it takes you to all the other bits.	4.2	
...to different screens...	5.2	5.3
...to do your work. Why do I work or why am I doing these? That's why I work. And I do what I'm told to do. Because otherwise I wouldn't be able to do my work...	6.2	7.4
Cause that's what I'm paid to do. Because I need the money.	7.2	

Table D.3: Content analysis of CU3

Response element	Category	Common categories
Just to show the system...	1.3	
...where to go... ...and then navigate,... I want to go to. ...which way to go.	2.3	1.1,3.2,5.4,2.5
...and which instruction to take. ...or another instruction.	3.3	
Because you need...	4.3	
...to go to another screen... To get to the screen...	5.3	5.2
Like if I want to input a bill,...	6.3	6.1
...I go to backslash...	7.3	
...invoice,...	8.3	
...standard,... ...and then standard.	9.3	
...well COMPANY first,...	10.3	
...to tell the computer...	11.3	

Table D.4: Content analysis of CU4

Response element	Category	Common categories
It leads you on... ...it takes you on to another route. ...that I need to go into.	1.4	
...to the next option...	2.4	
...that you require.	3.4	
It doesn't really stop there, does it?	4.4	
Well Navigate,...	5.4	1.1,3.2,2.3,2.5
I would need to do that... ...to actually physically get into the screen...	6.4	2.1
Obviously to help me... ...it doesn't enable me to carry out my tasks,... ...my day-to-day tasks,...my job. In my job they're obviously... I need to do it... Because I have to. ...this is part of my job... ...this is part of my day-to-day duties,... ...those are the tasks that I have to do. ...to do my work.	7.4	6.2
To input any invoices... ...to get these particular invoices... ...onto the system. ...the invoices... ...that need to be input to the system. Because without inputting these invoices,...	9.4	3.1,5.5
...to advise other people...	11.4	12.5
...of what's been issued to them,... ...that have been issued to their groups.	12.4	7.1

Table D.5: Content analysis of CU5

Response element	Category	Common categories
Find. I'd assume find.	1.5	
I know I've gone... ...to 'navigate'... ...as a system...	2.5	1.1,3.2,2.3,5.4
...but I've got no knowledge of it...	4.5	
Inputting some SYSTEM charges... To actually input the charges to... ...on our records. ...and record the necessary charges...	5.5	3.1
...on world-wide accounting.	6.5	
I understand they'd been ledgered in London in bar duty and we were obviously doing the opposite translation at our end.	7.5	
In order to keep our records in alignment... ...that had gone through.	8.5	
For actual accounting purposes... ...at month end.	10.5	14.1
To ultimately produce COMPANY's accounts.	11.5	
To provide information... To see how good we're doing... To prove that...	12.5	11.4
...to shareholders. ...as a business. ...we're a cost-effective concern... ...and we are making... ...a good profit for our shareholders.	13.5	

D.2.2 Academic User Group

Table D.6: Content analysis of AU1

Response element	Category	Common categories
It's for saving... ...you'd want to save...	1.1	1.2,2.3,1.4,1.5
...the contents of a file... ...to a specific filename... ...and file content. ...of the same file... ...of the same file...	2.1	2.3a
One to change the name... ...to a different name. ...different filenames. ...so if you want to make changes... ...and keep the old changes. ...what changes have been made.	3.1	4.2,16.3,2.4,3.5
And one to change the type of file... ...from its native format... ...to another format. ...that may not read the same format. ...because of different types...	4.1	2.2,2.5
...if you've got several different versions... ...and different versions...	5.1	6.4
...but want to keep separate copies of them... Other copies...	6.1	
...other people may be using...	7.1	12.2
...other kinds of software,... ...other packages...	8.1	7.2,8.4,5.5
...over time,...	9.1	
Version control.	10.1	
Historical record.	11.1	5.2,6.3
So you can keep your train of thought...	12.1	
So that you might not repeat yourself.	13.1	

Table D.7: Content analysis of AU2

Response element	Category	Common categories
That's for saving a file... ...with another,... ...of updating files... ...or if I want to save it...	1.2	1.1,2.3,1.4,1.5
...either format... ...as another format... I need different versions...	2.2	4.1,2.5
...or another name.	4.2	3.1,16.3,2.4,3.5
If I want to keep a record...	5.2	11.1,6.3
...that I can use... For using it,... I will use it for,... ...on the purpose of the file. ...or whatever the purpose of the file is. ...what the purpose of the original file is,... So the purpose... ...can be used	6.2	4.5
It is for my own purpose,... ...the purpose of later being able... ...or doing whatever I want to do with it. So that I can use it later on.	6.2a	5.3

...with another tool.... ...with a specific program... That depends on the application. Because I might have different applications... ...or different programs... I don't have one standard application... ...where it can be done. ...so that it can be done in different applications. ...if it's to run a program,... ...of the program... ...within other packages.	7.2	8.1,8.4,5.5
...a file can be only opened... ...to view the file... ...reading it... ...where the same file can be read from. ...or view it...	8.2	
If it is a text file then...	9.2	4.3
If it is code,...	10.2	
That's for my own convenience. To make it easier...	11.2	
...for other people to run it... So that others can use it...	12.2	7.1
...to run a simulation... ...or something else.	14.2	
...which has already been put down...	15.2	

Table D.8: Content analysis of AU3

Response element	Category	Common categories
Save as?	1.3	
...to save a file...	2.3	1.1,1.2,1.4,1.5
...to the local drive.	3.3	
Text files, it could be text files.	4.3	9.2
I can access it later on...	5.3	6.2a
...for future reference. For my information. Because it's for my information. ...and I need to store this information.	6.3	11.1,5.2
To print it.	7.3	
That's one of the options.	8.3	
Maybe to send it to someone...	9.3	
...if they have the facility.	10.3	
And because I cannot remember everything.	11.3	
I don't have to re-type everything. ...what I've just typed for instance,... ...or I want to write...	12.3	
That is one of the main problems,... ...points,...	13.3	
...why I would actually be saving it. ...a file or anything. ...or on the file itself...	2.3a	2.1
I don't want to repeat the same effort again,...	14.3	
I just want to load it...	15.3	

...and maybe update it... ...what kind of file. I need to update...	16.3	3.1.4.2.2.4.3.5
...a personal list,... ...a personal database of contacts,...	17.3	
...and I need to insert...	18.3	
...names,...	19.3	
...a letter... To send these sort of letters,... ...if we take the example of the letter,... ...to send these letters to my contacts.	20.3	
...that is a mailing list...	21.3	
Because I am using it,... ...I need to use it,... ...I'm going to make use of it.	22.3	
To receive a response from them...	23.3	
...if it's an invitation.	24.3	
They'll contact me.	25.3	

Table D.9: Content analysis of AU4

Response element	Category	Common categories
For saving a file... ...of the current file.	1.4	1.1,1.2,2.3,1.5
...with a different name...	2.4	3.1,4.2,16.3,3.5
...to the one you're [unclear].	3.4	
Because you do not want to overwrite.	4.4	
You want to create...	5.4	
...a different version... Sometimes because you want a different version...	6.4	5.1
Sometimes to make a backup copy.	7.4	
...because it's a slightly different application... ...you're developing...	8.4	8.1,7.2,5.5
...or a different direction...	9.4	
...of the document you're making.	10.4	
It depends on the situation...	11.4	

Table D.10: Content analysis of AU5

Response element	Cat-egory	Common categories
This is for saving a file,... ...or to save it... ...you need to save it...	1.5	1.1,1.2,2.3,1.4
...or a different format. ...or a different format. ...as a different format.	2.5	4.1,2.2
...probably with a different name...	3.5	3.1,4.2,16.3,2.4
If you want to use it...	4.5	6.2
...in another package... ...to use another package... ...they don't have this package... ...but they have a different package...	5.5	8.1,7.2,8.4
...in your 'A' drive...	6.5	
...to move the file to a different machine.	7.5	
Probably you want to continue working... ...somewhere else. Because you've finished working here. It depends on the place. ...to stay in this place. You probably start using this file in this computer. In the other computer... ...which you can use this time.	8.5	
You have no time...	9.5	
...which is not available.	10.5	

D.3 CSE content analysis

D.3.1 Commercial User Group

Table D.11: Content analysis of CU1

Response element	Category	Common categories
When you go into that particular screen.	1.1	5.3
You go into 'Navigate'...	2.1	2.2,9.3,2.5,4.4
...to input an invoice.	3.1	
That's the first button you press,...	4.1	1.2,6.4
...before going on to the next fields.	5.1	4.2,4.3
To input invoices.	6.1	7.3,10.4
We input invoices...	7.1	9.4,7.5
...to bill one part of COMPANY... ...to another part of COMPANY.	8.1	12.3,12.4,17.5
Because if one part of COMPANY is doing work... ...or providing services...	9.1	11.4
...to another part of COMPANY...	10.1	
...then they need to be charged for it.	11.1	
So that the books... ...but as correct as they possibly can be...	12.1	8.5
...at the end of the day are... I won't say one hundred percent correct... ...for each part of the business.	13.1	19.5
It's a requirement...	14.1	11.2,14.4
...under legislation...	15.1	

...and under Oftel...	16.1	
...that each part of COMPANY should be responsible...	17.1	
...for their own accounting purposes.	18.1	

Table D.12: Content analysis of CU2

Response element	**Cat-egory**	**Common categories**
...you can do anything with that. Just press...	1.2	4.1,6.4
...the 'navigate'... ...without using the 'navigate' button... ...in the first place.	2.2	2.1,9.3,2.5,4.4
...and then it takes you... So it'll get you...	3.2	
...to all the other bits.	4.2	5.1,4.3
So you can navigate around the whole system.	5.2	1.3,1.5
...to different screens...	6.2	13.3
...to do your work.	7.2	
Cause that's what I'm paid to do.	8.2	8.4a
Why do I work or why am I doing these?	9.2	
Because I need the money.	10.2	
And I do what I'm told to do.	11.2	14.1,14.4
Because otherwise I wouldn't be able to do my work	12.2	8.4

Table D.13: Content analysis of CU3

Response element	Cat-egory	Common categories
Just to show the system where to go... ...to tell the computer which way to go.	1.3	5.2,1.5
...and which instruction to take.	3.3	2.4
Because you need...	4.3	5.1,4.2
...to go to another screen...	5.3	1.1
...or another instruction.	6.3	
Like if I want to input a bill,... ...invoice,...	7.3	6.1,10.4
I go to backslash...	8.3	
...and then navigate,...	9.3	2.1,2.2,2.5,4.4
...standard,... ...and then standard.	11.3	
...well COMPANY first,...	12.3	8.1,12.4,17.5
To get to the screen...	13.3	6.2
I want to go to.	14.3	7.4

Table D.14: Content analysis of CU4

Response element	Category	Common categories
It leads you… …it takes you on to another route.	1.4	3.2
…on to the next option…	2.4	3.3
…that you require.	3.4	
Well Navigate,…	4.4	2.1,2.2,9.3,2.5
I would need to do that…	5.4	
…to actually physically get into the screen…	6.4	4.1,1.2
…that I need to go into.	7.4	14.3
Obviously to help me to do my work. …it doesn't enable me… …this is part of my job,… …this is part of my day-to-day duties,… …those are the tasks…	8.4	12.2
To input any invoices… Because without inputting these invoices,… …to get these particular invoices… …the invoices… …that have been issued…	9.4	7.1,7.5
…that need to be input to the system. …onto the system.	10.4	6.1,7.3
…to carry out my tasks,… …my day-to-day tasks, my job. In my job they're obviously…	8.4a	8.2
…to advise other people…	11.4	9.1
…of what's been issued to them,… …to their groups.	12.4	8.1,2.3,17.5
I need to do it… Because I have to. …that I have to do.	14.4	14.1,11.2

Table D.15: Content analysis of CU5

Response element	**Cat-egory**	**Common categories**
Find, I'd assume find.	1.5	5.2,1.3
I know I've gone to 'navigate'...	2.5	2.1,2.2,9.3,4.4
...but I've got no knowledge of it...	3.5	
...as a system as such.	4.5	
Inputting some SYSTEM charges...	5.5	
...on world-wide accounting.	6.5	
To actually input the charges to...	7.5	7.1,9.4
...on our records. In order to keep our records in alignment... ...and record the necessary charges... ...that had gone through. To prove that we're a cost-effective concern...	8.5	12.1
I understand they'd been ledgered in London...	9.5	
...in bar duty...	10.5	
...and we were obviously doing the opposite translation...	11.5	
...at our end.	12.5	
For actual accounting purposes...	15.5	18.1
...at month end....	16.5	
To ultimately produce COMANY's accounts.	17.5	8.1,12.3,12.4
To provide information to shareholders. ...for our shareholders.	18.5	
To see how good we're doing as a business.	19.5	13.1
...and we are making a good profit...	21.5	

D.3.2 Academic User Group

Table D.16: Content analysis of AU1

Response element	**Cat-egory**	**Common categories**
It's for saving the contents of a file…	1.1	1.2,1.3,1.4,1.5
…to a specific filename…	2.1	
…and file content.	3.1	
One to change the name… …to a different name.	4.1	4.2,3.5
And one to change the type of file…	5.1	3.2,6.4,2.5
…from its native format…	6.1	
…to another format.	7.1	6.2
…if you've got several different versions… …because of different types… …and different versions…	8.1	14.2,17.3a,8.4, 16.5
…of the same file… …of the same file…	9.1	18.3,8.4a
…but want to keep separate copies of them…	10.1	
…you'd want to use different filenames.	11.1	
…other people may be using…	12.1	24.2
…other kinds of software,… …other packages…	13.1	7.2,7.4,5.5
…that may not read the same format.	14.1	19.2
Other copies…	15.1	
…you'd want to save…	16.1	24.3,5.5a
…over time,…	17.1	14.5

...so if you want to make changes...	18.1	12.2,15.3
...and keep the old changes.	19.1	5.2a,3.4
Version control. So that you might not repeat yourself.	20.1	11.3
Historical record.	21.1	5.2,6.3
So you can keep your train of thought...	22.1	
...what changes have been made.	23.1	

Table D.17: Content analysis of AU2

Response element	Category	Common categories
That's for saving a file... ...or if I want to save it...	1.2	1.1,1.3,1.4,1.5
...with another,...	2.2	2.4
...either format...	3.2	5.1,6.4,2.5
...or another name.	4.2	4.1,3.5
If I want to keep a record...	5.2	21.1,6.3
...of updating files...	5.2a	19.1,3.4
...as another format...	6.2	7.1
...that I can use with another tool. ...a file can be only opened with a specific program... That depends on the application. Because I might have different applications... ...or different programs... ...so that it can be done in different applications. ...within other packages.	7.2	13.1,7.4,5.5
For using it, I mean.	8.2	14.3,4.5
It is for my own purpose,... ...the purpose of later...	10.2	
...being able to view the file... ...or view it...	11.2	
...or doing whatever I want to do with it. That's for my own convenience. So that I can use it later on.	12.2	18.1,15.3

It depends on the file. So if I don't have one standard application... ...where it can be done. That's why I need different versions...	14.2	8.1,17.3a,8.4, 16.5
If it is a text file then...	15.2	4.3
...I will use it for, reading it...	16.2	25.3
If it is code,...	17.2	
...where the same file can be read from.	19.2	14.1
It depends on the purpose of the file.	23.2	
...if it's... To make it easier for other people So that others can use it... ...can be used...	24.2	12.1
...to run it...	25.2	
...or whatever the purpose of the file is.	23.2	17.3,9.4
I don't know what the purpose of the original file is,...	25.2	
if it's to run a program,...	26.2	
...to run a simulation...	27.2	
...or something else.	28.2	19.3
So the purpose of the program...	29.2	
...which has already been put down...	30.2	

Table D.18: Content analysis of AU3

Response element	**Cat-egory**	**Common categories**
Save as? Yes, to save a file...	1.3	1.1,1.2,1.4,1.5
...to the local drive.	3.3	6.5
Text files, it could be text files.	4.3	15.2
Just to make sure...	5.3	
I can access it later on for future reference. For my information.	6.3	21.1,5.2
To print it.	7.3	
Maybe to send it to someone... ...a file or anything.	8.3	
...if they have the facility.	9.3	
And because I cannot remember everything.	10.3	
I don't have to re-type everything. I don't want to repeat the same effort... ...what I've just typed for instance,...	11.3	20.1
That is one of the main problems,... ...points,... ...why I would actually be saving it.	12.3	
I just want to load it...	14.3	8.2,4.5
...and maybe update it if I need to. I need to update...	15.3	18.1,12.2
Because it's for my information.	16.3	
It depends again...	17.3	23.2,9.4

...on the subject,...	17.3a	8.1,14.2,8.4, 16.5
...or on the file itself - what kind of file.	18.3	9.1,8.4a
...a personal list,... ...a personal database of contacts,... ...and I need to insert names,... ...or I want to write a letter... ...that is a mailing list... To send these sort of letters,... ...to send these letters...	19.3	28.2
...and I need to store this information.	24.3	16.1,5.5a
Because I am using it,... ...I need to use it,... ...I'm going to make use of it.	25.3	16.2
...to my contacts. To receive a response from them...	29.3	
...if it's an invitation.	30.3	
They'll contact me.	31.3	

Table D.19: Content analysis of AU4

Response element	Cat-egory	Common categories
For saving a file...	1.4	1.1,1.2,1.3,1.5
...with a different name to the one you're [unclear].	2.4	2.2
Because you do not want to overwrite.... You want to create a different version... ...of the current file.	3.4	19.1,5.2a
Sometimes to make a backup copy.	5.4	
Sometimes because you want a different version...	6.4	5.1,3.2,2.5
...because it's a slightly different application... ...you're developing...	7.4	13.1,7.2,5.5
...or a different direction...	8.4	8.1,14.2,17.3a, 16.5
...of the document you're making.	8.4a	9.1,18.3
It depends on the situation...	9.4	23.2,17.3

Table D.20: Content analysis of AU5

Response element	Category	Common categories
This is for saving a file,... Or probably you need to save it...	1.5	1.1,1.2,1.3,1.4
...or a different format.... ...as a different format.	2.5	5.1,3.2,6.4
...probably with a different name...	3.5	4.1,4.2
...or a different format. ...which you can use...	4.5	8.2,14.3
If you want to use it... You want to use another package...	5.5	13.1,7.2,7.4
...in another package...	5.5a	16.1,24.3
...or to save it in your 'A' drive...	6.5	3.3
...to move the file...	7.5	
...to a different machine.	8.5	
Probably you want to continue working...	9.5	
...somewhere else.	10.5	
Because you've finished...	11.5	
...working here.	12.5	
It depends on the place.	13.5	
You have no time... ...this time.	14.5	17.1
...to stay in this place.	15.5	
...which is not available. ...they don't have this package... ...but they have a different package...	16.5	8.1,14.2,17.3a, 8.4
You probably start using this file...	17.5	
...in this computer.	18.5	
In the other computer...	19.5	

Appendix E: Economic impact

E.1 Introduction

It might be felt that interface metaphors are simply the icing on the cake and that, although better metaphors might be a good idea for the individual user, they will have little outside impact. One obvious measure which would show whether such an impact exists is to assess the potential economic effects of metaphor choice. Chapter 4 described an experiment which looked at users' mental models of systems built with different categories of interface metaphor. Inspection of the users' descriptions showed that they described the system in terms of different layers of signification: the spatial metaphor concentrated the users' attention on the user interface, the activity-based metaphor on the system functionality, and the interactional metaphor on the tasks the system could support.

These changes in the users' perception are qualitative and are thus difficult to integrate with quantitative data such as economic performance. One tool which allows this type of integration is CRIMP (CRoss IMPact analysis tool), (Krauth 1994). Together with a colleague, I used CRIMP to test the potential techno-economic impact of metaphor choice which has been described in greater detail in a previously published paper (Condon 1995). The tests were based on an existing CRIMP model (Sinnigen 1994) of the techno-economic impact of advanced telecommunications services. The model was based on interviews and observations of organisations in a number of European economic sectors and was built by RACE project URSA (User Requirements and Strategies for Applications).

E.2 The CRIMP model

E.2.1 Cross impact analysis

CRIMP is a software package designed to simulate techno-economic systems by modelling the impacts of trends on other trends and on themselves. Trends can be anything which might affect or be affected by any other factor within the model. They can include non-quantitative considerations such as changes in technology or psychological factors, in addition to measurable factors such as costs and yields. Each trend is allocated an estimated value for the impact it has on the other trends and on itself. Once the model is built, the software steps through the changes in trends due to these impacts in a series of time frames. Each impact value is also allocated a confidence level which is used to randomly generate a series of models, varying each impact according to the associated level of confidence. For example, the impact of absenteeism on production levels can usually be predicted with a high degree of confidence and values might vary by no more than, say, 5% in any run-through. By contrast, the impact of absenteeism on general employee satisfaction would be difficult to predict accurately and values in different run-throughs might vary by as much as 50% from the initially predicted values. Actual values are not used by the model as each factor is normalised into a range from $-\infty$ to $+\infty$ with the initial value set to 0. This makes the tool valuable for modelling non-quantitative trends, as the modeller can judge a trend to have a 'small' or 'large' impact, setting appropriate levels of confidence.

Before using the model it is tested for internal consistency and stability by carrying out a series of individual randomised runs. If the model is inherently unstable, the predicted values will vary wildly from run to run. Sometimes this is a genuine property of the actual system under study, as when two incompatible formats, such as the VHS and Betamax video formats, are fighting for market dominance, and the outcome hangs on a few critical factors such as which manufacturers support which format. In other cases, the initial runs will vary only slightly as random variations are balanced by other factors. If a the model is found to be stable, then the experimenter can introduce 'actions' to the model to see how these affect its behaviour. Examples of actions might include the introduction of a fault into a product, an external event such as the entry of a new competitor into the market, or policy changes such as a price increase. The underlying principles and mathematics of CRIMP are described elsewhere. (Gordon 1968; Kane 1972; Helmer 1977; Duin 1995).

E.2.2 Factors

Four psychological factors were identified which could have a potential impact on usage trends. The first factor is *resistance* to new technologies. New services are rarely used extensively and to the full from the day that they are installed. This factor was identified in the URSA studies as a potential inhibitor:

> Human factor problems in the form of psychological resistance are often associated with process re-engineering or company re-organisation as it may be perceived by part of the management and the labour force as a threat to their position or to the control they exert. The old jobs consisted of specialists who did one task. The new case handlers perform a variety of tasks. Therefore people working on case handling process teams will find their work far different from the repetitious performance of one task to which they were accustomed. (Sinnigen 1994, pages un-numbered)

As explained in Chapter 4, activity-based interface metaphors turn the user's attention to the functionality of the services offered and, by making the user more aware of the functionality at their disposal, can help support the second factor, the general *usability* of the system. By providing interfaces which attract the user, spatial metaphors encourage initial *likeability*, the third factor. Finally, interactional metaphors tend to turn the user's attention towards the activities which the services support and therefore provide the final factor, the perceived *relevance* to the users' tasks. This might appear to point to interactional metaphors as the most suitable in the long term. However, arguments have been presented that a spatial metaphor such as the room is more suitable for real-time interaction in which the interaction itself is clear, whereas an interactional metaphor such as the form is better in supporting non real-time cooperation in which the interaction is less obvious (Hämmäinen 1991).

This gives four factors for the model which make up the input trends impacting on a fifth factor, *Usage*. The input trends do not have an even effect on usage over time and are therefore filtered through time series which change their impact over the time period for which the model is run. The profiles of the time series are described in Section E.2.2. The interactions between the trends and time series are shown in the diagram below.

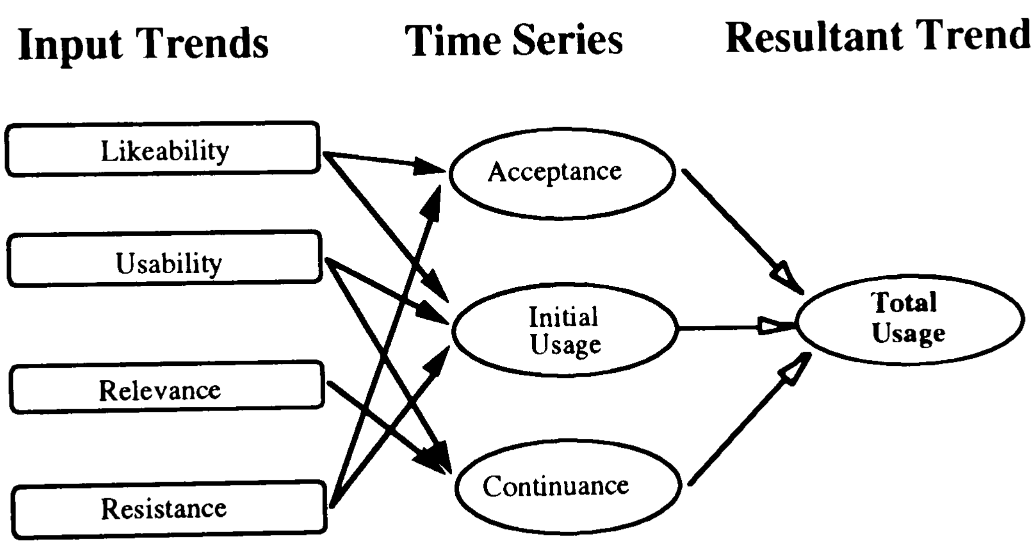


Figure E.1: Cross impacts in the MITS model

Duin (1994) provides a general guide to the figures which should be used when there is not a clearly measurable impact of one trend on another, as is the case with qualitative trends such as these:

0.1	->	Small Impact
0.5	->	Medium Impact
1.0	->	Large Impact
2.0	->	Very Large Impact

This results in the following matrix defined for the initial CRIMP model:

Table E.1 Cross-impact matrix

	Likeability	Usability	Relevance	Resistance	Usage
Likeability				-0.1	S
Usability					S
Relevance				-0.5	S
Resistance				0.2	-2
Usage					1

The figures show the impact of the factors shown in the side column on those listed at the top, where 'S' indicates a changing impact defined by a time series. It can be seen that likeability is defined as leading to a small reduction in resistance, and perceived relevance a medium one. However, resistance tends to build on itself, with a small-to-medium sized impact. All the trends also directly impact on usage. In the cases of likeability, usability and relevance, these impacts change over time, as described in the next section. Resistance has a constant, very large, negative impact on usage, while usage has a strong, positive impact on itself (as more people use telecommunications services, their usefulness increases). Although the scale of these impacts are based on 'common sense' rather than empirical evidence, run-throughs of the model with varying impact values showed no significant difference in final results.

E.2.3 Time series

In the cases of resistance and usage, the impact on usage is constant. For the other three trends, the manner in which they impact on usage was set as a time series with an emphasis on one of three time series: the *acceptance* of the services by the user, the *initial usage* of the services, and the longer-term *continuance* of use.

Service Acceptance. It is common to see some enthusiasts taking to new services immediately, with others taking a more cautious attitude. Although acceptance will be affected by all three trends, the dominant factor for the interface at this stage is likeability of the interface – whether the system looks attractive and interesting to use. A sufficiently likeable system will attract short term interest, even from users to whom it has no relevance and even if its usability is poor, as in the attraction of early VR demonstrations. This can be expressed as a very strong initial impact in favour of using the system which gradually fades away:

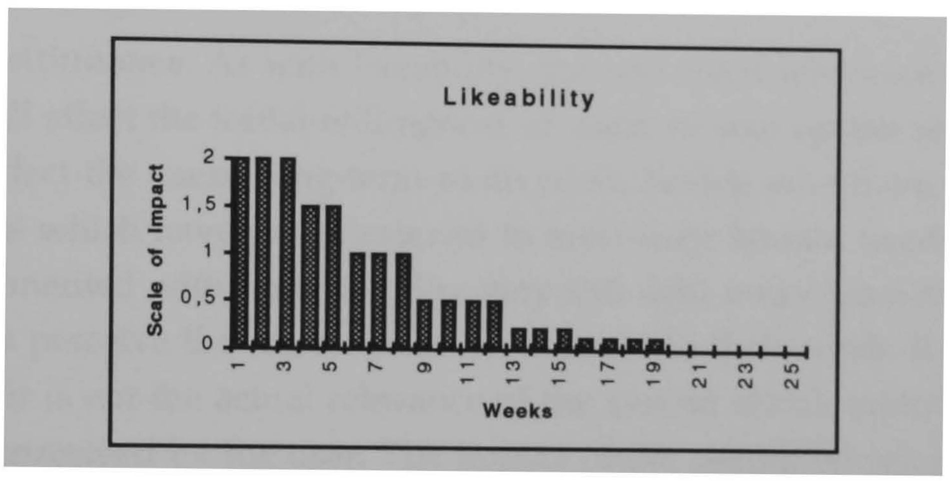


Figure E.2: Service acceptance time series

All time series are shown over a period of 25 weeks, after which it is assumed that usage will settle down at a relatively constant level. The scale of impacts is the same as that given in the main CRIMP matrix, i.e. from 0.1 (weak) to 2.0 (very strong).

Initial Usage. Poor usability may lead to a lack of usage even though the initial likeability of the interface attracted the user's attention and the system is perceived as having high relevance to the user's tasks. However, even with poor usability, some users will master the services and over time users will find themselves forced to use the system if it is essential for their work. Thus, poor usability will not effect immediate use nor long term use. However, good usability will help to speed user take-up of the system during the implementation phase, as shown in the graph below:

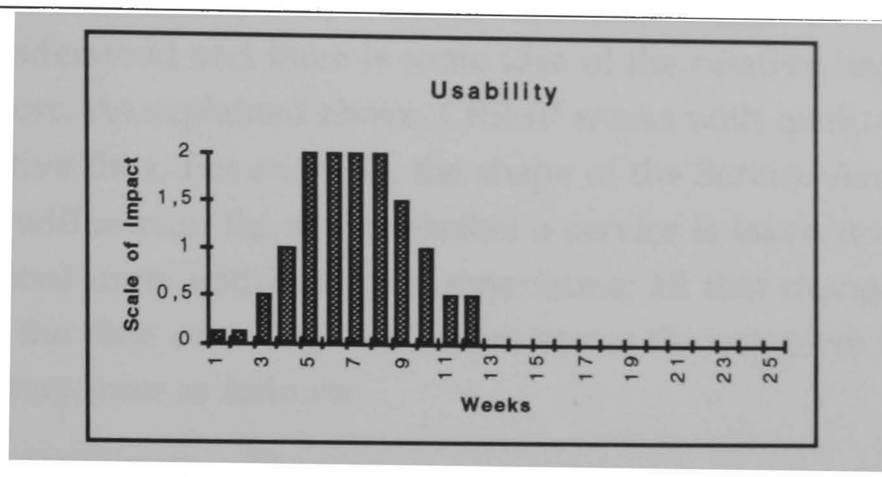


Figure E.3: initial usage time series

Service Continuance. As with likeability, the perceived relevance of the services will affect the initial willingness of users to take up the service but it will also affect the users' long-term motivation. People who have started to use systems which have been designed to meet their known needs and have been implemented with high usability may still drift away from their use if they do not perceive the relevance of the systems to their work. It should be noted that it is not the actual relevance of the system which matters, but the relevance perceived by the user. The impact of the perceived relevance of the system will initially be very strong. As factors such as usability involve the user more in issues of 'how' rather than 'why' to use the system, perceived relevance will be less important. Having mastered the usage of the system, the effects of motivation then come more clearly into play and will continue into the longer term:

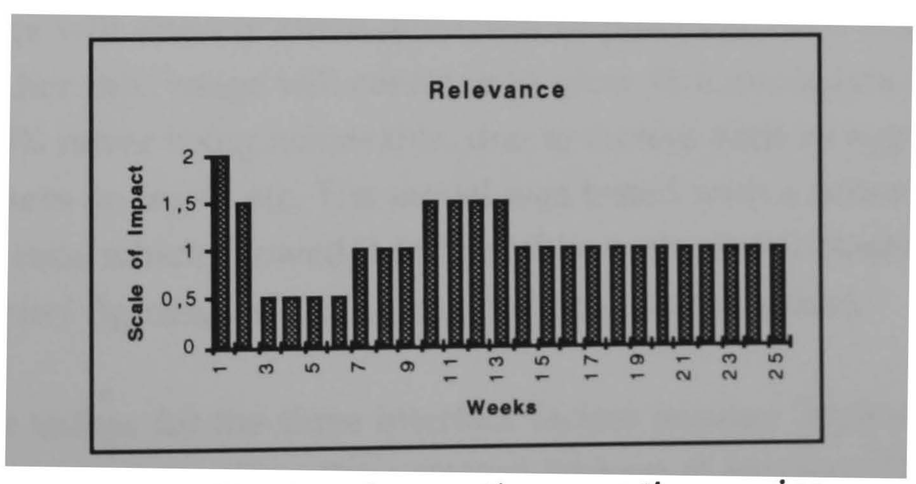


Figure E.4: Service continuance time series

In all cases, the exact formulæ for the graphs cannot be known. However, this does not affect the validity of the model as long as the **relative** time frames are understood and there is some idea of the **relative** impact of the different factors. As explained above, CRIMP works with qualitative data, not just quantitative data. For example, the shape of the Service Acceptance curve above will remain the same whether a service is taken up by 10% or 90% of potential users within a given time frame; all that changes will be the gradation of the time axis. The sum values across the complete for each of the three factors were as follows:

Service Acceptance	13
Initial Usage	13.2
Service Continuance	26.5

As can be seen, the sum for the Service Continuance time series is double that of either of the other two series. This was taken into account when defining the actions below.

E.2.4 Trends and actions

The four trends of likeability, usability, relevance and resistance were all given constant values throughout the time series of 50 on a scale of 0-100. This is an arbitrary figure representing 'typical' values for these factors at that point in the series: the actual figures for trends do not affect the CRIMP model – only changes to the figure have an impact. For overall usage it is expected that, typically, it will take time for users to learn to use the services but that usage will steadily climb, with 50% of potential users active within six weeks. After this, usage will continue to grow to a maximum of 85% – a figure of 100% never being achievable, due to factors such as equipment out of service, users on leave, etc. The model was tested with a series of randomised runs which showed it to be stable, with results closely correlated with the *a priori* figures, indicating a robust, consistent model.

The constant values for the three interface factors assume 'typical' interfaces, i.e. just good enough to allow the expected take-up of services. These could, in turn, be affected by poor interface design. Factors which have a single effect on a model are known in the CRIMP methodology as *actions*. Three actions were defined for the model: poor likeability (PoorL), poor usability (PoorU) and poor relevance (PoorR).

The model was run, in turn, with a negative impact for each of the actions on the relevant trend, e.g. with a cross impact of -10 of *PoorL* on *likeability*. As noted above, the total value of the Service Acceptance time series, which is used to filter the *relevance* trend, is twice as great as either of the others. To compensate for this, the impact for *relevance* was set at -5 (a 10% reduction on the 'standard' interface), whereas the impacts for the other trends were set to -10, a 20% reduction in the usability factor.

E.2.5 Results

By running the model with each of the actions in turn, the following usage profiles were obtained:

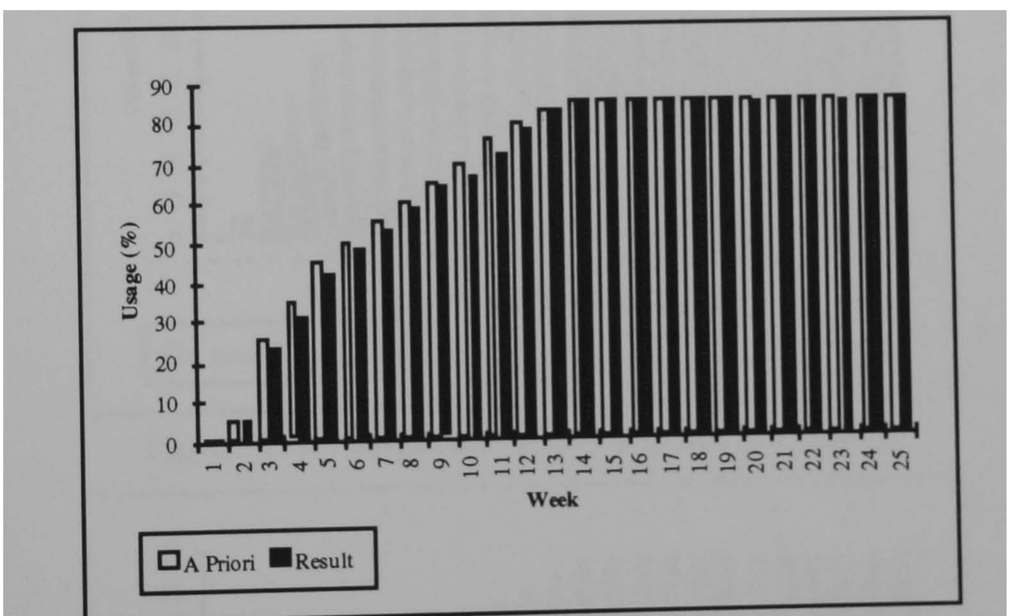


Figure E.5a: Impact of actions (initial values)

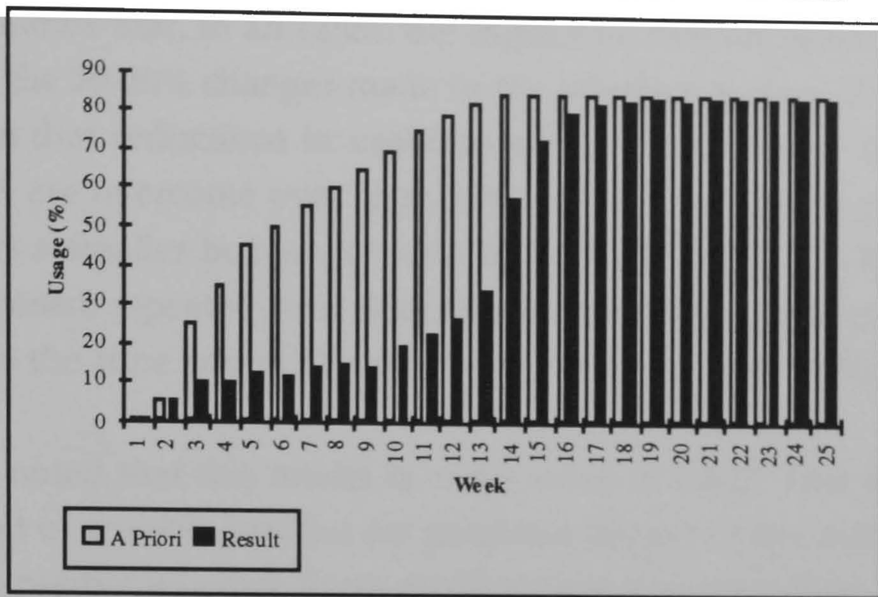


Figure E.5b: Impact of actions (poor likeability)

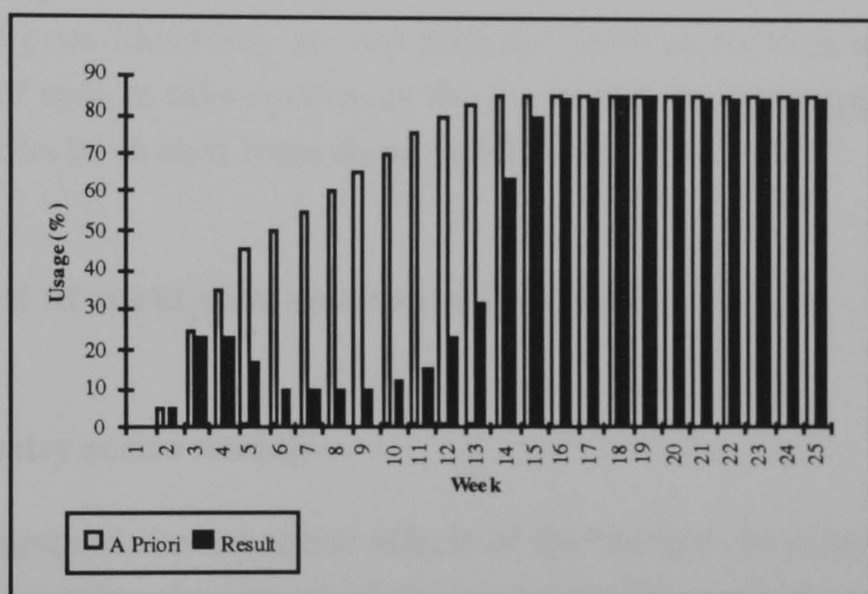


Figure E.5c: Impact of actions (poor usability)

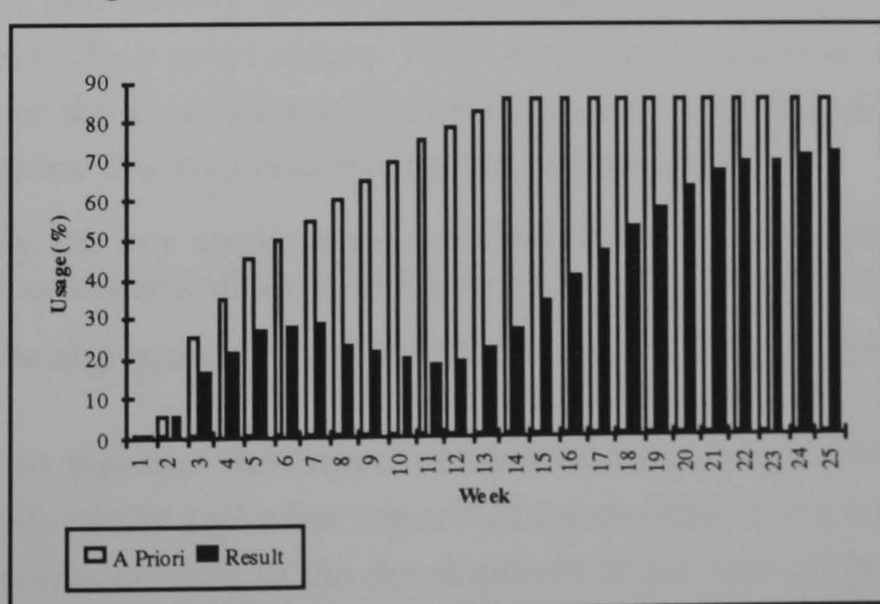


Figure E.5d: Impact of actions (poor relevance)

The output shows that, in all cases, the impact on overall usage is much greater than the 10-20% changes made to the interface factors. It also demonstrates that reductions in usability and likeability have immediate effects which are overcome over time, whereas a reduction in perceived relevance has a smaller but longer term impact. Although the figures were arbitrarily chosen, repeated runs with smaller and larger impacts, and with variations on the time series, showed very similar overall profiles.

It should be noted that this model is of no value in itself. This model merely provides a set of scalable profiles for potential impact of the metaphor choice – it does not predict whether those profiles have any impact on organisational performance. In practice, differences in the profiles for poor usability and poor likeability are not relevant. Both show dips in usage in the early stages of system take-up but, as the initial figures were approximate, no conclusions can be drawn from these small differences.

E.3 Impact of metaphor choice on industry

E.3.1 Industry sector models

To fully understand the economic effects of the changes in usage, it is necessary to examine the impact of the usage profiles within specific industry sectors. URSA had already carried out a number of such studies which were used to provide the *a priori* values. These resulted from a systematic and in-depth study of the innovative usage of advanced communications in many economic sectors. The two objectives of this study were:

- To identify the key applications on which innovative demand for advanced communications is likely to be based in the European economy
- To describe and quantify the benefits generated by these applications

It is difficult to measure the impact of advanced communications on company productivity and other aspects of competitive advantage, as advanced services are still in the development phase and usage conditions in pilot experiments cannot often be compared with real commercial usage conditions. URSA therefore adopted an in-depth simulation based case-study approach to measuring user benefits, complemented by an empirical survey. This approach involved the following steps:

Identify key applications for sectors into which economic activity is aggregated at the EU level.

Reconstruct key value generating processes of representative companies in the different sectors.

Simulate impacts of the identified applications on the outcome of the value generating processes in each company.

Check validity of simulation results in an empirical survey on company acceptance of the identified applications. A summary of simulation results was presented to 120 companies distributed across the sectors, and their feedback was collected through interviews and questionnaires.

Benefits in terms of productivity gains together with expected penetration rates allowed for the calculation of a monetary equivalent of the total impact of the identified applications on each sector. Data on turnover, employment and number of companies was taken from European Commission surveys (Eurostat 1993). The full details of URSA's work can be found in the relevant project deliverable (Sinnigen 1994).

E.3.2 Method

The usage profiles from the initial usage model were put into selected models from the URSA project, chosen to accord with the economic sectors for which the MILAN system had been designed and those used as pilots by the MITS project. Although the initial model appears to show a dominant effect for the relevance factor, this is not necessarily the case when the data is combined with other effects. For example, likeability has a much greater impact in the very short term which could, in some cases, be more important than the longer term effect of perceived relevance.

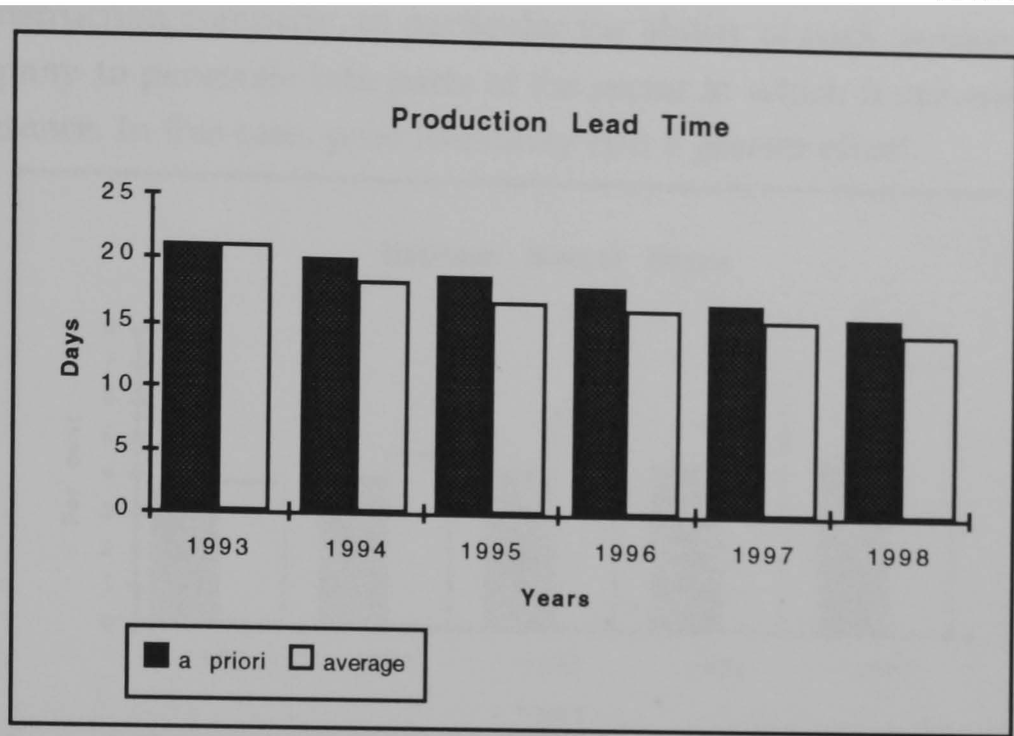
Two types of action to be applied to the sector models were therefore defined:

Delay: the impact of poor likeability and usability in the early stages can be summed up as a delay in take-up.

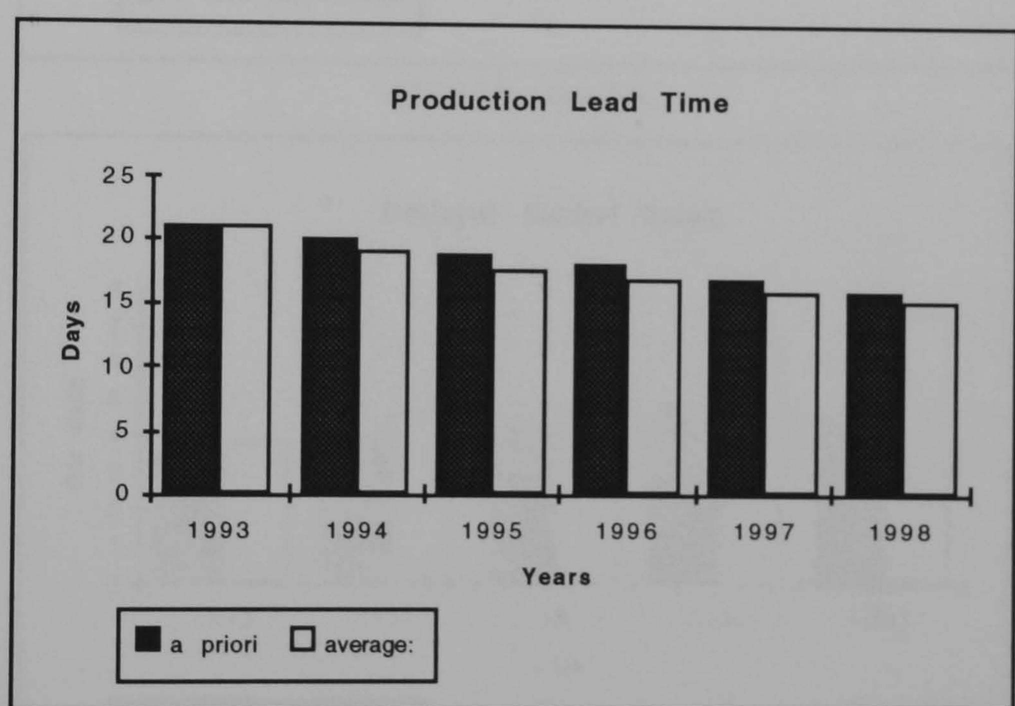
Reduction: although not as dramatic in its initial impact, poor relevance leads to a small longer-term reduction in usage.

E.3.3 Electrical engineering

The chosen application examined in the electrical engineering model was that of EDI (Electronic Data Interchange), in this case referring to the exchange of information with customers and suppliers, including videoconferencing as well as more conventional EDI. The results of the unmodified URSA model showed a reduction in lead time of 1.5 days as advanced services were implemented. Reductions in likeability reduced this only in the first year, but poor relevance reduced the long term improvement by one third (from 1.5 days saved to 1 day).



URSA's results

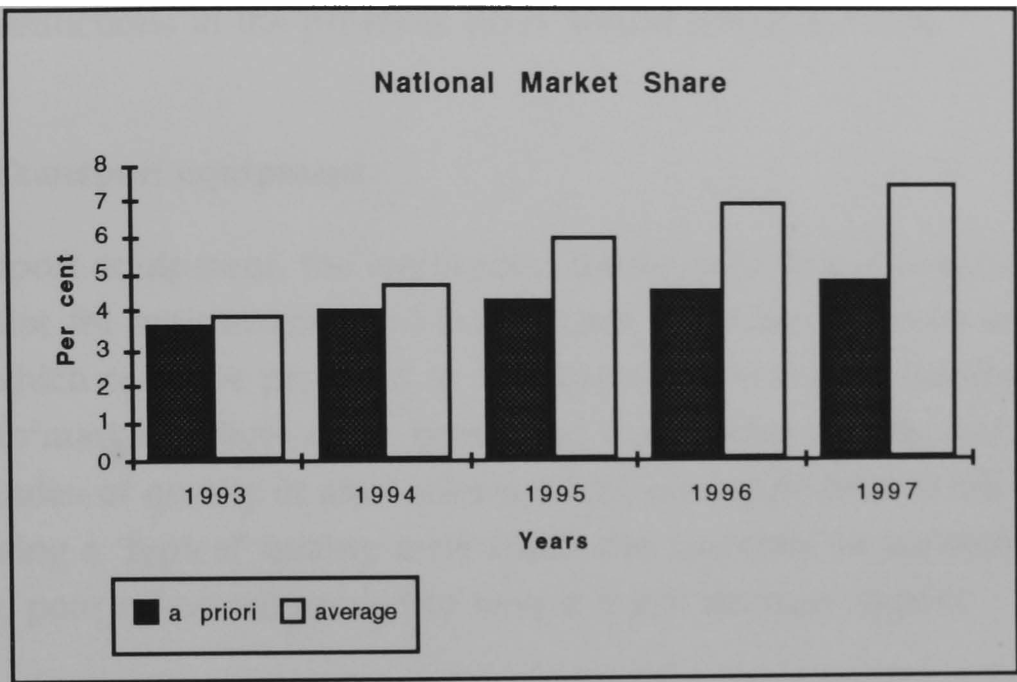


Impact of poor relevance

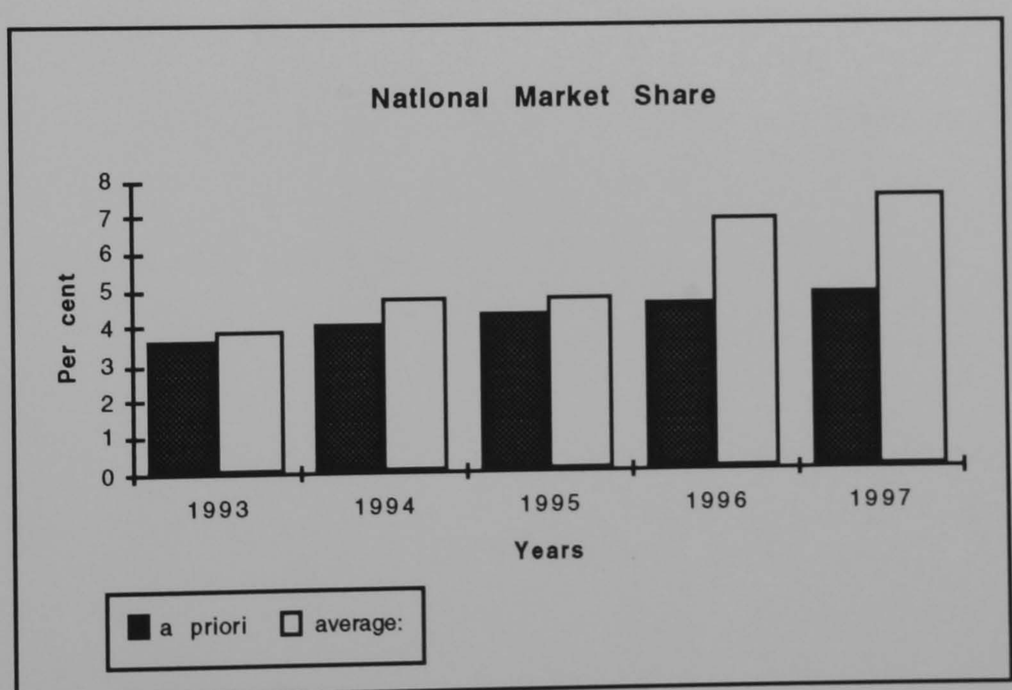
Figure E.6: Impact of poor relevance in electrical engineering

E.3.4 Construction

For construction, the chosen application was more dramatic and more obviously linked with metaphor: the virtual meeting room. URSA related this application to its potential impact on the national market share of a large construction company, in particular the ability of such services to allow the company to penetrate into parts of the sector in which it currently has little presence. In this case, poor likeability had a greater effect.



URSA's results



Impact of poor likeability

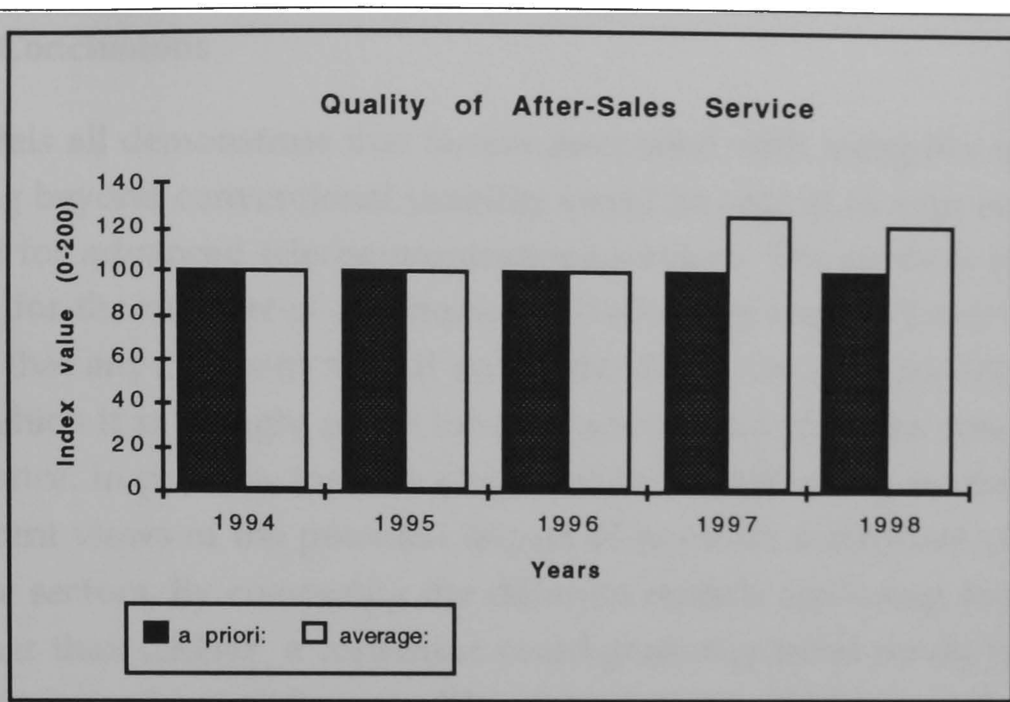
Figure E.7: Impact of poor likeability in construction

In this case, the impact of the new services is predicted as dramatic and steadily increasing, both in absolute and proportional terms. By the final year of the model, the potential market share for the company becomes 7.45%

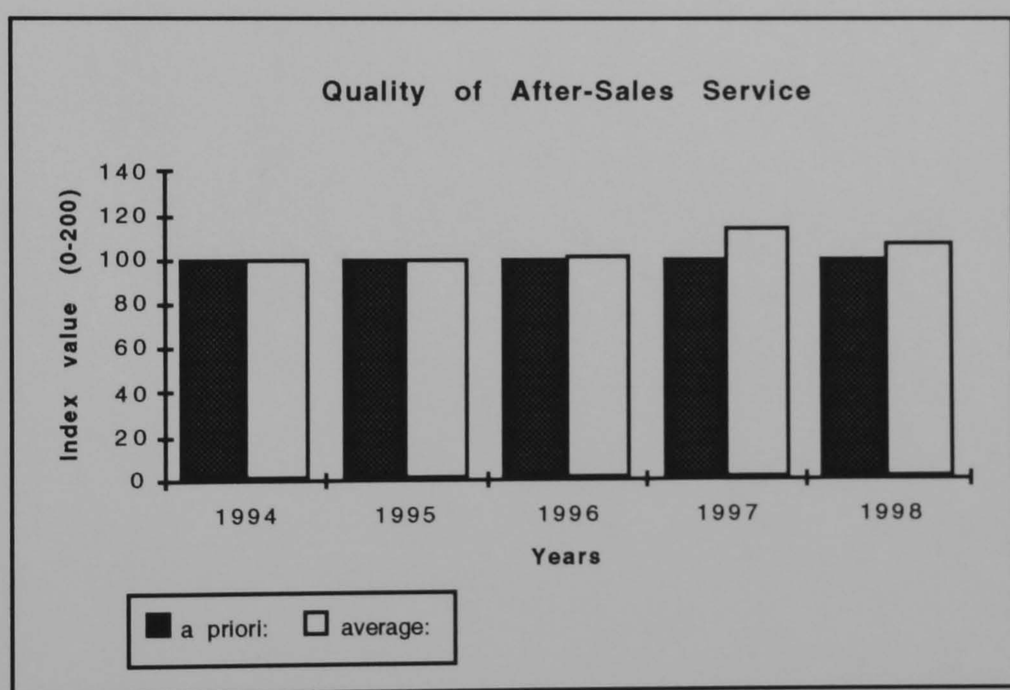
rather than 4.8%. The impact of poor likeability continues to have a significant impact for three years after its introduction. Although the potential market share in the final year appears to be little reduced, it is likely that the reductions in the previous years would jeopardise this.

E.3.5 Transport equipment

For transport equipment, the application chosen was that of remote delivery of expertise for maintenance and interference rectification. As an additional service which could be provided to customers, this was not related by URSA directly to market indices of the types used in the other models, but to a general index of quality of after-sales services, on a scale from 0-200 with 100 representing a 'typical' quality level which can currently be maintained. In this case, poor relevance proved to have a much stronger impact:



URSA's results



Impact of poor relevance

Figure E.8: Impact of poor relevance in transport

As a more advanced service which will not be available for some time, its impact does not come into effect until near the end of the time series. The impact of this service is therefore not as strong as those of the applications examined in the other sectors. However, by the end of the time series the impact of the new service with poor relevance factors has faded away almost completely, bringing the quality level back to a value of 107.

E.3.6 Conclusions

The models all demonstrate that factors associated with metaphor usage and extending beyond conventional usability could be critical in supporting the potential for advanced telecommunications services. The study is open to criticism for the number of assumptions which were made. However, it is unlikely that any company would accept the deliberate introduction of factors which it is thought might have an adverse effect on the company's performance. In practice, the next step must be to build many models based on different views of the potential impact of interface metaphors on various economic sectors. By comparing the different models and using modelling tools other than CRIMP, a consensus could gradually build up on which factors were model-specific and which represent true effects.