Authors' Details

Department of Information Systems and Computing,
Brunel University, West London, Uxbridge, Middlesex UB8 3PH, UK
Email: {Chris.Condon, Mark.Perry, Bob.Okeefe}@brunel.ac.uk

Telephone

020 8993 4051 (home)

01895 274 000 (Brunel)

Fax: 01895 251 686

Bionotes

Chris Condon is a Research Fellow specialising in human-computer interaction in the Department of Information Systems & Computing at Brunel University. After gaining degrees in ergonomics and computer science, he worked in industry in Britain and Germany before gaining a PhD in the application of semiotics to HCI and joining Brunel.

Mark Perry lectures in human-computer interaction in the Department of Information Systems & Computing at Brunel University. He has degrees in psychology and cognitive science in addition to PhD work in the area of HCI. He was previously a Visiting Scholar at Stanford University and a Research Fellow at Brunel.

Bob O'Keefe is Professor of Information Management in the Department of Information Systems & Computing at Brunel University. He is also associated with Henley Management College. He is the co-editor of the European Journal of Information Systems, and was the Research Track Chair of the 14th Bled Electronic Commerce Conference.

Denotation and connotation in the human-computer interface: The 'Save as...' command.

Chris Condon, Mark Perry and Robert O'Keefe

Department of Information Systems and Computing,

Brunel University, West London, Uxbridge, Middlesex UB8 3PH, UK

***Email: {Chris.Condon, Mark.Perry, Bob.Okeefe**}@brunel.ac.uk*

Keywords

HCI, semiotics, interface design, metaphor, signification

Abstract

This paper presents a semiotic technique as a means of exploring meaning and understanding in interface design and use. This is examined through a study of the interaction between the "file" metaphor and "save as" command metaphor. The behaviour of these (from a functional or computational basis) do not exactly match, or map onto, the meaning of the metaphor. We examine both the denotation of a term to the user, i.e. its literal meaning to that person, and the term's connotations, i.e. any other meanings associated with the term. We suggest that the technique applied is useful in predicting future problems with understanding the use of metaphor at the interface and with designing appropriate signification for human-computer interaction. Variation in connotation was expected but a more fundamental difference in denotation was also uncovered. Moreover, the results clearly demonstrate that consistency in the denotation of a term is critical in achieving a good user understanding of the command.

1. Introduction

Every part of a computer system that a user interacts with conveys some form of meaning to that user - intended or not. In semiotics, anything that carries a meaning is known as a sign, c.f. Saussure (1974), whether it is a word, a picture or the overall 'look-and-feel' of something. A sign signifies a number of things at once: its *denotation* (usually its literal meaning) and a number of *connotations* which will depend on its context and vary from person to person. Following de Saussure's terminology, with additions by Eco (1979), the perceived sign is known as the *signifier*. What it signifies is then known as its *signified*, and the process of associating one to the other its *signification*. Adapted from Eco (1979, p. 58-59), the combination of these three concepts forms the *sign-function*, as illustrated in the diagram below (fig. 1):

Signifier	**Signification**	**Signified**
	Recognition that the Signifier relates to the Signified.	File container.
The Sign, perceived in context as the 'folder' icon.	Associating the perceived Sign with the Signified.	What the 'meaning' of the Signifier is considered to be.

Figure 1: The sign-function.

It is central to effective communication that, in a given context, a sign must hold a similar denotation for both the person creating the sign and the person interpreting it. Even the denotation of a physical sign will be dependent on context to some extent. For example, the word 'chip' has a quite different denotation when used in the context of computing from that which it has when discussing food, even though the two meanings share the same etymology and are thus listed as the same word in the dictionary. This should be distinguished from variations in connotation which are far more strongly dependent on context and on individual interpretation. A picture of an aircraft on a motorway sign denotes an airport but its connotations in the minds of specific individuals could include an air crash, a holiday in the sun, a boring business trip or an unfocused fear of flying.

Many signs in computing are based on metaphors which bring their original denotations and connotations with them, creating a potential for confusion in the user's understanding. This paper briefly examines the role of metaphor in HCI, and the fact that many different metaphors can co-exist

in a single computer. As a first stage, semiotics is offered as an approach for analysing potential metaphors with an aim to identifying problems they might introduce. To illustrate this, the use of the 'Save as...' command in Microsoft Word was examined using a semiotic technique. As expected, many different connotations were uncovered. More surprisingly, the work also showed that widely used computing terms can lack a clear denotation. There are problems for any interface that carries inappropriate connotations for the user, but confusion over the denotation of a standard computing term could carry much more serious consequences for the design of all computing systems. This paper concludes that one reason for this lies in the fact that many computing terms are formed from metaphors.

2. Metaphor in HCI

2.1 On metaphor

Metaphors are of particular interest, as they can bring their original denotations and connotations with them, creating a potential for confusion in the user's understanding. Semioticians regard metaphor and metonymy (naming something after one of its components or attributes) as central to language: see Eco (1985) or Lodge (1990) for a wider discussion than is possible here. Eco singles out metaphor as particularly important, "The majority of our messages, in everyday life or in academic philosophy, are lined with metaphors." (Eco 1985, p.262), while Lakoff and Johnson (1980) claim that metaphor is central not only to our language but also to our thought processes. In terms of the evolution of our language, the field of computing is very recent and introduces many novel concepts for which we require new terminology. Occasionally we create new words, for example through acronyms such as 'wysiwyg', but metaphor remains the dominant technique for naming new concepts in computing (Johnson 1994).

Although a metaphor will initially be recognised as such by users, the new use of the term will take on a separate meaning of its own and people's perception of it as a metaphor will gradually die. 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). In semiotic terms, this means that a separate denotation has formed. For example, if a new type of computer utility were to be called a 'hare' because it speeds up operations, users would apply the current denotation of a long-eared animal to the program and recognise it as a metaphor, related to the hare's reputation for speed. Over time, a reference to the computer 'hare' would be recognised as a separate sign, a new *signifier* with a quite

separate denotation, just as a computer 'mouse' now carries a quite separate denotation from that of the animal, at least to those people familiar with computing.

The components of a metaphor consist of where it is taken from, known as the *vehicle*, what it applies to (the *tenor*), the factors shared between the two, known as the *ground*, and the effort demanded to match one to the other (the *tension*) (Richards 1936). Applying this to the hypothetical 'hare' program gives the following:

Table 1: Components of a metaphor.

Vehicle	Ground	Tenor	Tension
Hare (fast, small furry animal).	Speed.	Utility to speed up computer operations.	Small furry animal v. software program.

Many researchers in the past have emphasised the value of metaphor to computing, particularly as part of the learning process:

> 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 *et al.* 1988, p.67).

Way (1991, p.8) 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." However, researchers have recognised problems when the metaphor vehicle does not fit the tenor (Carroll and Thomas, 1982), whilst another study by Carroll and Mack (1984) looked 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 this case the metaphor was not even intended by the interface designers, demonstrating that if designers do not employ metaphors then users will often adopt their own.

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." (Dieberger and Tromp 1994, p.57). For example, when using the cut-and-paste metaphor, we are not limited to pasting a clipping once but can paste it repeatedly, an example of what Kay (1990, p.199) and others have termed the 'magic' added to the metaphor.

Some have argued that metaphors should not be used at all, but their arguments are generally in favour of choosing a different type of metaphor. For example, Halasz and Moran (1982) suggest that algorithmic approaches are more useful, but the only example they give suggests the better applicability of the 'move' command to change a directory 'tree' rather than using the desktop metaphor, although both the 'move' command and the 'tree' structure were originally metaphors. More recent attacks have come from Nelson (1990) who commends spreadsheets and suggests using hypertext instead of metaphors, and Kay (1990) who offers agents as an alternative. Again, spreadsheets and agents are both metaphors, while hypertext systems are dominated by the use of metaphors (see below).

2.2 Choice of metaphor

Metaphors are of particular importance when examining the signification of the interface to the user. While metaphors may help a user to understand the new and unfamiliar by reference to the known and familiar, every metaphor also brings its own set of significations to add to the multiple significations of the concept it represents. 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 (Smith *et al*. 1982). 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 and providing a 'physical' environment rather than a language of interaction (Smith *et al*. 1982, p.246-247).

The desktop metaphor has become the standard interface for almost all personal computers, although the purity of the original concept has been eroded through the introduction of unrelated metaphors, e.g.

the Windows Explorer. Whilst adding power and functionality to the interaction, the value of these 'composite' metaphors (Carroll *et al*., 1988) has been debated with respect to their usability. The desktop metaphor is also challenged by the growth of home computing and the Internet, and the consequent introduction of new metaphors which metaphorically break out of the office such as 'browsing' or 'surfing' the web and 'visiting home pages'. We can also question the appropriateness of the 'office' metaphor in a leisure environment.

Hypertext introduced its own metaphors such as 'hyperbooks' and the 'cards' of HyperCard. Its combination with multimedia on the Internet has now led to a series of communications or link-based metaphors such as 'net' and 'web'. Other 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), information fields and swamps (Bernstein 1993), information forests (Rifas 1994), or urban metaphors such as the city (Dieberger 1994). A separate, and commonly used, category of metaphor is that of the interface agent or guide to show the user around (Laurel 1990, 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 'paper clip' Office Assistant).

Many HCI researchers have examined interface metaphors but there is there is no generally accepted consensus on which metaphor is most appropriate in a particular circumstance - each use of a metaphor must be researched individually. Carroll's work, like that of most early researchers, used techniques from experimental psychology, which proved useful in showing the benefits or drawbacks of specific metaphors, but did not give a general model of the operation of the metaphor process. Other researchers have offered cognitive approaches to provide potential ways to model the use of metaphor, such as those based on problem spaces (Douglas and Moran 1983) or mental models (Norman and Draper 1986, Fischer 1991). For example, the fit of a metaphor, and therefore its appropriateness, can be seen as the degree to which the user's mental models of the tenor and vehicle overlap. An experiment based on this approach (Anderson *et al.* 1994) showed that users form more accurate models of a system when the 'conceptual baggage' - i.e. the number of features in the metaphor vehicle which are not present in the system - is at a minimum. However, when the experiment was repeated with more complex interfaces by Condon and Keuneke (1994) it was found that users could

make very effective use of metaphor-based interfaces without forming coherent mental models of them (see also Tognazzini, 1992). One interpretation is that the users built action-oriented mental models (Young 1983) as these do not have to be coherent, but without this property it is difficult to see how the mental model approach can be used to model the process of metaphor.

2.3 Categories of metaphor

The most important issue in metaphor design for HCI is that of finding suitable metaphor vehicles but it is worth considering the many different tenors to which they can be applied in computing. Metaphors are obviously used for technical terms such as files, records, windows and menus, but also for design concepts such as objects and dataflow. Different metaphor vehicles often co-exist on a single computer or at different stages of development without attracting widespread criticism. For example the original metaphor that inspired the Apple Macintosh was that of seeing a computer as a household appliance like a toaster (Carlton 1999). This bears little relation to many of the metaphors used in the development of the architecture, such as objects and inheritance, or those used in the Macintosh interface. As an example, table 2 shows some of the metaphors that can co-exist in a recent version of the Macintosh, the Apple iBook:

Table 2: Metaphors in the Apple iBook.

Metaphor type	**Implementation**
Concept	Appliance
Development	Objects, classes, inheritance
Hardware	Book
System	Windows, menus
Presentation	Desktop
Interaction	Direct manipulation
Networking	Port, web, chat room
Support	Balloon help

This may seem like a confusing mix of metaphors, such as those criticised by Carroll (see above) but user interaction is usually through a single metaphor at a time. The computer can be carried, opened and closed as a book, but the user can forget this when interacting with the desktop items. It is possible to drag a document to a folder but not to a menu, and clicking on the document to enter an application or access the web will introduce new metaphors but will normally mean leaving the desktop environment behind. Generally, each metaphor belongs to a different aspect of human-computer interaction, such as the object-oriented metaphors which were used in the design process but are invisible to the end-user and thus cannot interfere with understanding of the desktop metaphor.

In summary, experimental approaches such as the studies by Carroll *et al.* described above can give valuable insight into the problems and advantages of specific metaphors but provide no general model of the role of metaphor in computing. By contrast, cognitive approaches offer a promising coherent model but the experimental evidence shows that users can use a system consistently well despite forming an inconsistent and inaccurate model of it. What is required is an approach which allows the building of a model which allows for this inconsistency. Semiotics appears to offer just such an approach. It is a central tenet of semiotics that a sign will hold many significations for a user which will often contradict one another. The most basic of these, the denotation, needs to be consistent across users, but each user will also be aware of many connotations of a sign which will vary between individuals and contexts. On some occasions the connotations of a sign might alienate the user, as has particularly been suggested by feminist theorists (Grundy 1996) but, provided the user is willing to use the system, the basic level of usability is not directly affected. If, however, there is confusion about the denotation then this raises serious questions about the basic usability of the system.

3. Denotation and connotation - a study of metaphor in use

3.1 Semiotic techniques in HCI

Semiotic techniques provide an alternative approach to the experimental and cognitive approaches discussed above. Semiotics has already been applied to the analysis of a wide range of communication forms - Blonsky (1985) gives examples from advertising, television, cinema and political posters - and it should be equally applicable to examining communication with computers. Semiotic approaches have already been used to explore some aspects of interface design but these have rarely concentrated on the signification of the interface to the user. The signification of signs is explored by sections 1 (*Signification and Communication*) and 2 (*Theory of Codes*) of Eco's *Theory of Semiotics* (Eco 1979,

p.32-150). This work is one of the most important studies in semiotics and has been used by Andersen in developing his *Theory of Computer Semiotics* (Andersen 1990), and by Stamper (2001, also Stamper *et al.* 2000) who has used semiotics as a basis for describing the structure of information systems. Eco's work has provided one of the sources for research in 'Semiotic Engineering' (de Souza 1993, Prates *et al.* 1997, Pimenta *et al.* 1997). However, their work derives from the third part of Eco's book, his *Theory of Sign Production* (Eco 1979, p.151-313), seeing the human computer interface as a means by which system designers create signs to communicate with users. Although this can provide important insight into the design process, it cannot be assumed that users will recognise the same signification as intended by the designer. Of greater relevance to this paper is the work of Liu *et al.* (1998) who successfully developed and employed Andersen's approach for the analysis of user interfaces. He used structured questionnaires to break down users' opinions of three different interfaces. In contrast with Liu's reductive approach, this paper employs a semiotically-derived technique to 'open out' the signification of a sign.

A central feature of Eco's *Theory of Codes*, also expressed by Barthes (1973), is the recursive nature of signification. The awareness that a particular signifier led to a particular signified can lead to a further level of signification. For example, a standard denotation of a smile is 'a sign indicating happiness'. However, in a particular instance the first signification that might come to mind could be very different, such as "what is he after?" Having raised this new signification, we might then ask why we are being so suspicious, the feeling of suspiciousness acting as a new internalised sign leading to a new signification and so forth. Generally, if we ask a user what an interface feature or computing function is for, the reply will be based on a single signification of the sign to the user. As the work described below shows, this will usually be the denotation but this is not necessarily the case. To gain access to the recursive levels of signification of a sign, we must question the user in a similarly recursive manner. To do this, a simple technique has been proposed: the 'What for?' interview, in which a user is presented with a sign and asked "What is that for?" (Condon and O'Keefe 2001). When the interviewee answers, the interviewer replies by again asking, 'What for?', (or occasionally 'Why?' if that is the easier form to express). In this way, it should be possible to steadily uncover many of the connotations of the sign to the user by repeatedly questioning the basis of their understanding. The results generated from this will allow the researcher to gain greater knowledge of what the sign means to the user.

3.2 The interviews

The interview technique was applied to two groups of people using Microsoft Word. One element was chosen from the interface to form the starting point of the questioning: the 'Save as...' command on the pull-down 'File' menu. The 'save' metaphor is now dead for many users, certainly for the experienced users in this study, and has been included as a computing term in dictionaries for over decade, such as the entry in Chambers (Schwarz 1988), "to store (data) on a tape or disc." The first user group was composed of five postgraduate researchers working in a number of areas of information systems and computing, principally information modelling. Given that the study lies in the field of HCI, researchers from this field were excluded from this user group. The second group, who could be expected to be less familiar with the technical workings of the program, but possibly more experienced in using it, were administrative staff from the same department. The intention was to minimise differences in results due to environmental or organisational factors as both groups work in the same building within the same overall technical and management structure. The main distinctions between the groups lie in the type of work they do and the fact that the two groups socialise separately, each discussing their work with their own colleagues in their own language.

It is not sensible to hope that the technique could uncover every possible signification for all users – some will be non-verbal and users might find it impossible to describe them, others will simply not come to mind at that particular moment. As such, the study constituted an 'exploratory case study' (Robson 1993), rather than an experiment. In considering the sample size, and in contrast to experimental studies, a single subject is able to provide a great deal of useful information about their understanding of the system, whilst extending the interview group to a hundred people would not necessarily add much more. Instead of using statistical sampling, *purposive* sampling was used (Robson 1993 p. 155-56): in this case, the purpose of the interviews being to consider whether the technique could be useful to interface designers. It is not suggested that the technique would provide all the information that a designer needs but that it should be one of the tools available for user requirements gathering. In a pilot study carried out prior to the one documented here, the design of bespoke systems was considered - most of which were used by four or five users. Whilst generic applications might be used by a much larger number, initial usability studies by the designer are likely to be limited to a similar scale. The central research question thus becomes, "Can the technique deliver important information for an interface designer when applied to small groups of users?" 'Important'

information is considered to be anything which might have a significant impact on the design of the system.

The study was intended to establish the denotation and connotations of the interface within a work context and all interviews were therefore carried out as close to the interviewees' normal work situation as possible. Interviews with all subjects were undertaken by the first author. 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 some research 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 questions towards the end of the interview. In addition to the standard assurances of anonymity, the interviewer also read out the following statement before each interview:

> "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."

The first step of the study procedure was to point out the "Save as..." command 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.

3.3 Results

The interviews were very short, the shortest taking one minute seven seconds, the longest three minutes 37 seconds, demonstrating that the responses formed a spontaneous expression of the users' views of the system. As the length of the interviews (table 3) shows, data collection for this type of study can be an extremely quick process.

Table 3: Interview length for administrators and researchers.

Length of Interview		
Administrators	Min	1 min 7 sec
	Max	1 min 57 sec
	Average	1 min 27 sec
Researchers	Min	0 min 58 sec
	Max	3 min 37 sec
	Average	2 min 5 sec

Content analysis techniques were used to break down the responses into separate significations (see Condon and O'Keefe [2001] for further details). Two analysts independently performed the analysis by breaking down each response into separate fragments at each point where a potentially new signification was introduced. As there were no pre-defined categories, each of the analysts used the separate significations they had identified as categories for cross comparison, forming 'extensional lists' (Krippendorf 1980, p.76-77). The process obviously contains a subjective element, as it is up to the analyst to identify whether two fragments carry the same meaning. The results of the two analyses were therefore compared using the standard test for Kendall's Tau with ties in conditions, showing significant correlation between the two sets of data at $p<0.05$. The tables below therefore show the arithmetic averages of the two sets of analyses. Table 4 summarises the numbers of separate significations:

Table 4: Summarised results of the study.

Number of Significations		
Administrators	Min	9
	Max	20
	Average	13.5
Researchers	Min	10

	Max	25
	Average	17.6

To summarise the important figures here, the data show an average of 17.6 separate significations for the researchers' interviews and 13.5 for those of the administrators. The difference in the averages is probably due to the different type of work undertaken by the two groups (see below) although the number of subjects is too small to measure any statistical significance in the figures. However, of particular interest with respect to this paper are the terms used by the interviewees to refer to certain aspects of the system. The terms 'file' and 'directory' in the top row of table 5 below refer to their standard meanings as used in the Word manual, while the term 'filing' refers to the practice of organising files according to a system. The terms shown in the second row of table 5 were the terms used by the interviewees to refer to these concepts:

Table 5: Terms used by administrators and researchers.

Technical term	**File**			**Directory**		**Filing**
Users' term	File	Document	Other*	File	Folder	Filing/ on file
Admin						
A		1	4*	1		2
B		3		1	3	2
C	1	2	3**	1		1
D		1				
E			1**			
Total	1	7	8	3	3	5
Research						
F	6					
G	11		1**			

H	5					
J	2	1				
K	3					
Total	27	1	1	0	0	0
*The user used the term 'report' to refer to a file. **The users used the term 'record' to refer to a file .						

It can be seen from the table that the two groups of users tended to use very different terms to refer to files, with most of the administrators avoiding the term 'file' itself while almost all the researchers used no other term. Researchers did not talk about directories or the filing process at all, although almost all of the administrators did. This should not be taken to imply that these were the only references to saving information, as most of the interviewees also used terms such as 'saving something' or 'saving your work' where reference was to the contents of the file rather than the file as an entity. It should be noted that content analysis is, to a degree, subjective and that some comments are ambiguous. For example, at one point administrator C used the phrase '... so I've still got a record on file.' This was interpreted as using the word 'record' to refer to a file but might equally well be taken as a reference to the general practice of file keeping rather than the specific file. The term 'on file' was taken as referring to the practice of filing, although it could have been intended as a reference to a directory, given that other administrators used the term 'file' in this manner.

In a previous study, interviewees had raised a number of issues of potential importance to systems design and management, most notably confusion among users about the ultimate purpose of the system (Condon and O'Keefe 2001). However, the responses of the study described here are of more relevance to human-computer interaction and interface design. Examination of the interviews shows that the 'Save as...' command was used in a very different manner by the two groups, with very different connotations, including some that could lead to problems. More importantly, the interviews showed that the term 'file' held not only different connotations but very different denotations for the two groups.

All ten users identified the denotation of the term 'save' in response to the initial 'what for?', using phrases such as, "It's for saving the contents of a file," or, "...save it on your disc." However, the distinction in denotation between the 'Save as' and the 'Save' command was expressed differently for

the two groups. Four of the five researchers said they use the command to change the file type of a file, while three of them said they used it to change a file's name; none of the administrators mentioned either of these significations. In contrast, all but one of the administrators mentioned using the command to file a document or put it into a particular directory, a signification mentioned by none of the researchers.

The two different denotations reflect the different working practices of the two groups. It is clear from the full interviews that the researchers usually created documents for their own purposes whereas the administrators created documents for other people who often came back with requests for changes or corrections, or asked for new copies of the printed document. The administrators had strict routines for naming files and filing systems in which, for example, documents created for a particular person were always saved in a particular folder. By contrast, the researchers used the command in a less organised way for a wider variety of reasons, creating additional versions of a file, saving copies in other formats to be opened by other applications, or creating a back up (all administrators' machines were automatically backed up). However, each set of responses uncovered confusion with the denotation of computing terms that could lead to problems.

4. Discussion

4.1 The 'file metaphor'

The word 'file' was previously applied to a collection of records physically collected together, usually in an organised way, which was used as a metaphor to describe the organisation of electronic data. In traditional data processing work the analogy provided a good fit. For example, Kilgannon (1980, p. 74) provides an illustration in which computer files are compared with the folders in a filing cabinet. With the move to personal computing, the term has been generally applied to the 'files' used on PCs, even when these do not contain discrete records. 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 Macintosh uses the terms 'document' and 'item' to describe entities on the desktop, the 'File' menu is standard for Macintosh applications. Similarly, Microsoft products use the term 'Documents' in the 'Start' menu but refer to them as 'files' in the 'File' menu and the 'File Manager'.

In contrast to the researchers who worked with computer files most of the time, the administrators spent a lot of their time dealing with physical files which are more closely equivalent to a 'folder' or

'directory' on a PC. The administrators made four references to 'filing' as an activity and five references to 'files'. However, only one of these was an explicit reference to a computer file in the phrase, "It's for saving a file". The other four references were more ambiguous but appeared to be references to computer folders rather than files. For example, one administrator referred to saving a document 'under a particular file or folder' where the terms 'file' and 'folder' appear to be used interchangeably. Rather than using the term 'file', administrators preferred to use other terms, such as 'document', 'record' and 'report' using these on sixteen occasions, contrasting with only three references to them by the researchers.

It was to be expected that the term 'file' would hold different connotations for the two groups. In an office environment, files are associated with the practice of filing, whereas most of the researchers are practised computer programmers who would be more likely to associate files with file handling and file conversion. It has already been noted that the administrators made many references to filing. The main connotations associated with this were related to organisation and clarity, with all of the administrators referring to the ease of retrieving files, making comments such as 'so it's not all haphazard', or 'it makes it clearer for me'. Such connotations were not mentioned by any of the researchers apart from one reference to 'version control', although the researcher did not explain what he meant by this.

Despite the difference in understanding between the two groups, there was no suggestion that this led to problems in using files, even among the administrators who used the term inappropriately. One reason for this could be the fact that computer files form part of the extended metaphor of the desktop. Physical files are understood in the context of other physical items in the office, such as the shelves and cabinets where they are kept and the desktop or table on which they are opened. Computer files are understood in the context of the other virtual items in the interface, such as the windows and the folders where they appear to be kept and the applications and menus used to manipulate them. As was stated at the beginning of the paper, signification is contextual, even in the case of the denotation. Thus it may be that an administrator will use the term 'file' to refer to a directory when speaking about the system to another person but then switch to a different linguistic code when actually using the system.

4.2 The 'Save as' command

In contrast to the variety of terms used to refer to a file by the administrators, as shown in table 5 above, files were referred to as 'files' by all of the researchers, with 28 references to 'file' or 'files'

and only one to 'document', while none of the researchers used the term 'file' to refer to a directory. It is important to be clear as to what the 'Save as...' command actually does. When chosen, it prompts the user to choose a name, location and file type for saving the current contents of the open window, defaulting to the current file name, location and file type. Despite the researchers' responses, the command 'Save as…' does not change the name or type of a file. If a new name or file type is chosen then the contents of the window are saved in a separate file without overwriting the original. This confusion about the functionality of the command could come from users' confusion as to what a file is: whether an identical copy made to another drive is still the same file, whether a file remains the same file if it saved under another name or file type, or if its contents are changed.

Microsoft appear to have recognised the problems with the 'Save as' command in more recent versions of Word. In Word 5.1 for the Macintosh, if a file is opened, amended, then saved as text, all formatting is lost, both in the copy saved to disc and in the open window. The 'save' metaphor implies that information will be saved, not lost, but it is not possible to undo the operation. In the more recent Word 98 for Windows NT, some effort has been made to correct this problem. With this version, a warning is given that formatting might be lost, together with an option to abandon the operation. Despite this warning, no information is actually lost until the document in the open window is closed. It is therefore quite possible to ignore the warning, with the intention of saving the file later, and then forget to do so.

It is possible that the problem could have been dealt with by the use of a different command from 'Save as...'. One candidate might be the 'export' metaphor which has been used by some other programs. For example, Adobe PhotoShop 2.0 had 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. The importance of this is that JPEG compression results in the permanent loss of graphical information. Because the image is being exported to 'JPEG territory', the 'Export' command cannot be used to over-write the existing file and lose information. In PhotoShop 2.0 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 2.0 does not treat the changes

as having been preserved. The application 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 metaphors used here appear to more closely match the actual behaviours exhibited by the system.

The behaviour of PhotoShop 2.0, with its separate 'Export' and 'Save as...' commands can be compared with that of Word, which attempts to use the same command 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, the user can even overwrite the original Word file and lose not only the format changes but all pre-existing formatting information. The pressure to standardise, with the domination of Microsoft as the *de facto* standard, means that PhotoShop 5.0 has dropped the 'Export' command. In the newer version it is possible to 'save' an image in JPEG format, over-writing the original file and destroying information without even a warning. However, PhotoShop 5.0 does have an additional 'Save a Copy...' command. This behaves identically to 'Save as...' in all but two respects. Instead of defaulting to the file name, the command defaults to the filename followed by the word 'copy', making it less easy to overwrite the existing file. More importantly, the program does not mark the file as saved if the 'Save a Copy...' command is used, even if the existing file has been overwritten. Saving a copy is, in fact, closer to what the 'Save as...' command is meant to do and might have been a more logical name for the command.

5. Conclusion

The 'Save as...' command and the file metaphor are now so standardised that it is probably too late to change them, although use of 'Export...' or 'Save a Copy....' might help to prevent the loss of data, as might a more complete implementation of the 'Undo' command to allow users to correct the matter. However, it is not too late to help prevent similar problems in future interfaces. The 'What for?' technique is easy to use and takes very little time, augmenting rather than replacing current methods for identifying user requirements. The results of this short study suggest that it could be extremely useful in predicting future problems with interface signification.

When this research began, it was expected that usability problems might exist due to the connotations of a sign. For example, a number of writers have attacked the use of male gendered metaphors as excluding female users, such as Grundy (1996, p.88-92) who attacks the use of 'rape metaphors', giving examples such as 'violation', 'degradation', 'chaining', 'abort', 'kill' and 'execute'. In the case

of 'file' and 'save' the problems do not lie in the connotations of the words but in their denotation. Whether this distinction matters in terms of a system's usability is arguable - Fiske (1982, p.92) sees the ability to distinguish denotation and connotation as one of the "main aims of semiotic analysis", but Hall (1980, p.133) writes that the terms are "merely useful analytic tools," and "must not to be confused with distinctions in the real world." However, it is denotation that lies at the heart of our language. Although a term might have many different connotations to different people, it is the similarity of the denotation that allows them to communicate. If one person refers to a dog as a 'dog' while another calls it a 'banana' then they will find it difficult to talk about dogs without establishing a new code of interpretation. When the term 'file' applies to a physical object, such as a box file, the denotation is clear - different files are physically distinct from one another. When the term applies to a computer file, the distinction is far less clear.

This paper was written using Word. It has been emailed between the authors who have both made copies on their own machines and converted it from one version of Word to another. Copies of old versions have been saved under different names and the two systems backed up at various times. Is it reasonable to regard these as different files or as different copies or versions of the same file? With physical files or data processing systems there would be no confusion. For example, people might refer to the 'payroll file' as a single logical entity, even though it exists in separate physical versions for each month's pay, with additional copies as back-up. However, the relationship between these copies and versions will be strictly controlled with an established naming convention that identifies each one as a version of the file. These constraints do not apply on a PC and it is difficult to see where the boundary of a 'file' lies.

Without stepping back through time it is impossible to assess the precise degree to which its origin as a metaphor was to blame for users' confusion over the meaning of the term 'file' demonstrated in this study, but the above account does suggest reasons why it may do so. These concerns are not nit-picking over the use of words: such incorrect signification can lead to real usability problems, particularly when combined with the signification of the term 'save'. The problems also challenge the most common basis for selecting metaphors. It appears to make sense that a computer interface used in an office environment should use metaphors taken from that environment. However, this underlying assumption has never been convincingly justified - this research suggests that metaphors might be better taken from a different environment to avoid confusion between the components of the physical

office and those of the computer. As end-user computing is undergoing continuous and rapid change, the meanings of computer terms can evolve separately. The result is that a word's denotation as a computing term can be at odds with its denotation when applied to a physical object. For the administrators in this study there is clearly confusion over what a 'file' is which might not exist if a less familiar metaphor had been chosen.

Saussure labelled semiotics 'synchronic', which he defined as "concerned with the logical and psychological relations that bind together coexisting terms and form a system in the collective mind of speakers." (Saussure 1974, p.99-100). This approach contrasted with the then prevalent 'diachronic' approaches which concentrated on the history of language and the roots of individual words. This research has generally followed a synchronic approach and demonstrated confusion in users' current understanding of some terms but the investigation of the reasons for this confusion would require a diachronic approach, tracing or reconstructing the history of individual terms over time. The speed of change in computing is such that the term 'file' never has time to settle to a single unambiguous denotation before it becomes as redundant as 'core' memory, a term which originally referred to a specific way of storing data, metaphorically extended and confused with any form of RAM, but now largely forgotten. The growth of the Internet can only add to this type of problem, as more and more features of computing become networked or 'virtual'. Unlike a physical object, it could be argued that a concept such as the Web has no existence until it is named, so that the denotation is inevitably unclear. A single Web page might already be split across locations spread around the world, while individual users can give themselves many virtual personalities. When designing interfaces for such an environment it is essential that we understand the terms we use and that we understand what they signify to users, designing for consistency of denotation at the very least.

References

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

Anderson, B., Smyth M., Knott R., Berger J. and Alty, J., 1994, Minimising conceptual baggage: making choices about metaphor. In: *People and Computers IX, HCI `94.* Glasgow, August 1994, pp. 179-194.

Barthes, R., 1973 [1957, in French], Myth today In: R. Barthes, *Mythologies* (London: Collins/ Paladin).

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

Blonsky, M., (ed.) 1985, *On signs* (Oxford: Basil Blackwell Ltd).

Carlton J., 1999, *APPLE: The inside story of intrigue, egomania and business blunders* (New York: Random House).

Carroll, J. M. and Thomas, J. C.,1982, Metaphor and the cognitive representation of computing systems. *IEEE Transactions on Systems, Man and Cybernetics.* **12** April 1982.

Carroll, J. M. and Mack, R. L., 1984, Learning to use a word processor: by doing, by thinking and by knowing. In: J. C. Thomas and M. L. Schneider, (eds), *Human factors in computer systems* (New Jersey: Ablex).

Carroll, J. M., Mack, R. L. and Kellogg, W. A., 1988, Interface metaphors and user interface design. In: M. Helander, (ed.) *Handbook of human-computer interaction* (Amsterdam: North-Holland).

Condon, C. and Keuneke S., 1994, Metaphors and Layers of Signification: the consequences for advanced user service interfaces. In: H-J. Kluger, A. Mullery, and N. Niebert (eds), *Towards a Pan-European Telecommunication Service Infrastructure - IS&N '94* (Springer-Verlag, Berlin), pp. 75-88.

Condon, C. and O'Keefe, R., 2001 The User's View and the User's Voice: A Semiotic Technique. *Information Systems Journal* (submitted for publication).

de Souza, C. S., 1993, The semiotic engineering of user interface languages. *International Journal of Man-Machine Studies*. **39**, 753-773.

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

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

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

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

Eco, U., 1985, The semantics of metaphor. In: R. E. Innis, (ed.) *Semiotics: an introductory reader* (Bloomington, Indiana: Indiana University Press).

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

Fiske, J., 1982, *Introduction to Communication Studies* (London: Routledge).

Florin, F., 1990, Information landscapes. In: S. Ambron and K. Hooper, (eds) *Learning with interactive multimedia* (Redmond, Washington: Microsoft Press), pp. 27-49.

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

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

Halasz, F., and Moran, T. P., 1982, Analogy considered harmful. In Proceedings of the ACM Conference on Human Factors in Computer Systems (Gaithersburg, Md., March 15--17 1982) pp. 383--386.

Hall, S., 1980, Encoding/decoding. In: Centre for Contemporary Cultural Studies (ed.) *Culture, media, language* (London: Hutchinson).

Johnson, G. J., 1994, Of metaphor and the difficulty of computer discourse. *Communications of the ACM*. **37** (12).

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

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

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

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

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

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

Liu, K., Crum, G. and Dines, K., 1998, Design issues in a semiotic description of user responses to three interfaces. *Behaviour & Information Technology*. **17**, 175-184.

Lodge, D., 1990, Narration with words. In: H. Barlow, C. Blakemore and M. Weston-Smith, (eds) *Images and understanding* (Cambridge University Press)

Nelson, T. H., 1990, The right way to think about software design. In: B. Laurel and S. J. Mountford, (eds) *The art of human computer interface design* (New York: Addison-Wesley Inc.).

Norman, D. A., and Draper, S. W. (eds), 1986, *User centred system design: new perspectives on human-computer interaction* (Hillsdale, N-J: Lawrence Erlbaum Associates)

Pimenta, M. S. and Faust, R., 1997, HCI and Requirements Engineering - Eliciting Interactive Systems Requirements in a Language-Centred User-Designer Collaboration: A Semiotic Approach. *SIGCHI Bulletin*. **29**.

Prates, R. O., de Souza, C. S. and Garcia, A. C. B. G., 1997, A semiotic framework for multi-user interfaces. *SIGCHI Bulletin*. **29**, 28-39.

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., 1993, *Real world research: a resource for social scientists and practitioner-researchers* (Oxford: Blackwell)

Saussure, F. de, 1974 [1915 in French], *Course in general linguistics* (London: Fontana/Collins).

Schwarz, C., (editor-in-chief), 1988, *Chambers English Dictionary* (Edinburgh: W & R Chambers Ltd.).

Smith, D. C., Irby, C., Kimball, R., Verplank B. and Harslem, E., 1982, Designing the Star user interface. In: *Byte*. **7** 242-282.

Stamper R. K., Liu, K., Hafkamp M. and Ades Y., 2000, Understanding the role of signs and norms in organisations: a semiotic approach to information systems design. *Behaviour & Information Technology*. **19** 15-27.

Stamper R. K., 2001, Organisational Semiotics: Informatics without the computer? In: K. Liu, R. J. Clarke, P. B. Andersen and R. K. Stamper (eds), *Information, Organisation and Technology* (Boston: Kluwer Academic Publishers), pp. 115-171.

Tognazzini, B., 1992, *Tog on Interface* (Reading Mass.: Addison-Wesley)

Way, E. C., 1991, *Metaphor and knowledge representation* (Boston: Kluwer Academic Publishers).

Young, R. M., 1983, Surrogates and mappings: two kinds of conceptual models for interactive devices. In: D. Gentner and A. L. Stevens, *Mental Models* (London: Lawrence-Erlbaum).