Assess The Relevance Of Metaphors To Human – Computer Interaction From The Perspective Of The Design Essay, Research Paper

System metaphors appear to be attracting more and more attention and you often hear the term banded about in journals and conference proceedings. The aim of this essay is to examine the potential usefulness of metaphors to the users and designers of systems, concentrating, in particular, on the human-computer interface. It will hopefully be established whether system metaphors are phenomena worthy of serious thought or whether it is simply an empty term with no more than intuitive appeal which will be passed over and forgotten with time. The basic idea behind a metaphor is to understand a new concept in terms of one which is already familiar and understood. A well known metaphor is that of Rutherford’s comparison of the hydrogen atom to the solar system. Lakoff and Johnson (1980) see language to be structured metaphorically and claim that this reflects the structure of thought processes in general: Metaphor is pervasive in everyday life, not just in language but in thought and action (Lakoff & Johnson 1980, p3) Many teachers have used metaphor and analogy to facilitate their pupils learning and indeed analogy seems to be a central aspect of learning. Metaphors can be distinguished from models by the partial nature of the correspondence between the old and new concept. What is then the relevance of metaphor to human-computer interaction? A significant interest in HCI is controlling system complexity. This affects the learnability and ease of use of a system and consequently its acceptance by users. If a system is not accepted by users then in many ways it is a failure. There are different approaches to reducing system complexity and improving learnability and usability: firstly the number of actions the user has to execute and the number of processes and concepts to be remembered can be reduced; alternatively, instead of reducing the initial complexity designers could increase the initial familiarity of a system by using interface actions, procedures and concepts which make use of specific existing knowledge and experience that users have of other domains. This is the broadened meaning of the term metaphor which is employed in HCI research. Metaphors can be of use for both the user and the designer, since what is good for the one is frequently good for the other. The user’s behaviour at the interface is often believed to be shaped by the mental model thet he or she has of a system and task. The knowledge that it represents enables an expert user to predict and understand a complex system. Metaphors can be regarded as short-cuts along the path to acquiring a comprehensive mental model. They can be used to lend a familiar air to unfamiliar circumstances which might otherwise prove confusing. Gentner (1980) speculates that the difference between experts and novices can be explained in terms of analogies: …the expert has an abstract global model with broad scope, while the novice has a pastiche of rich, only locally useful models…naive models of science appear more like expressive analogies than do expert models There is evidence that even when not provided with a metaphor users spontaeously create them anyway as part of the understanding process (Carroll & Thomas 1982). It would be useful if designers could anticipate likely metaphors in order to guide users towards useful ways of thinking about a system and prevent them from selecting inappropriate metaphors which would perhaps hinder learning. As Carroll, Mack & Kellog (1987) point out “the use of interface metaphors is pervasive and growing in interface design”, however it is not clear to what extent designers are aware that they are using metaphors. At present it seems to be a rather intuitive and haphazard process. A metaphor which has appeal for a designer may not necessarly convey the correct concepts to a user and may have broader implications which are not superficially evident. These and many other factors seem to imply that user interface designers should learn all they can about what metaphors are, how they have effect and how they can be incorporated into user interface and system design in a systematic manner. “”"”"”"”"”"”"”"”"”"”"”"”" This next section aims to present some of the main metaphors which are presently used, as well as discussing more experimental metaphors. The metaphor of a typewriter is frequently used to help people master word processing systems. Much useful knowledge about keyboards and the act of typing itself can be applied to word processing, reducing the complexity of the system and making it easier to learn. Whiteboard and Chalkboard are both systems which exploit prior knowledge about the very same real-world objects as well as attributes of these objects such as a common work space for cooperative groups, and freeform text and graphics. Other frequently used metaphors include document and business forms, but perhaps the best known metaphor is that of the desktop, a complex composite metaphor. Commercial examples of this metaphor include the Xerox Star system as well as Apple’s Lisa and Macintosh systems. This metaphor is intended to enable users to manipulate familiar office objects in familiar ways and exploits people’s ability to use the space around them for organising and storing things. For example, a document folder can be retrieved from a filing cabinet. Documents can be represented as icons and simultanious activities in different places as stacks of paper, as on a real desk top. Files that are nolonger needed can be thrown away in the rubbish bin. The physical world metaphor of direct manipulation is frequently used for pointing and selecting different objects and tools. Desktop tools embody add-ons to the desktop metaphor. They are small, separate applications which are similar to office tools such as notepads, calenders and calculators. Physical world or real world metaphors usually involve real world objects and their attributes. For example Boxer, an object-oriented programming environment, uses boxes and spatial properties, such as containment, to represent semantic properties of programming constructs. More generally these can be viewed as belonging to the spatial metaphor which uses ideas related to visual representation and direct manipulation, two closely related concepts. Visualization can be a powerful aid to comprehension and conceptualization as demonstrated by the use of graphics and even animation. Navigation through an information space is being investigated as an effective means of retrieving information and several systems have been developed around this concept. Bannon et al (1983) developed a multiple virtual workspace intended to help users organise windows effectively. Each workspace, called a Room, holds a collection of windows related to different primary tasks. Users can move between Rooms by selecting the appropriate Door icon. Users are described as being able to navigate their way through the system and tailor room contents to their needs. To me this suggests the different rooms of a house which contain different functional objects, for example, a kitchen with a cooker, a fridge, etc. Many hypertext systems present some means for displaying and navigating through a network structure. All of these systems incorporating spatial metaphors raise questions. For example, how can non-numerical information be positioned in information spaces? How can concepts be mapped onto a two or three dimensional view? How can we provide users with navigational aids to help them find what they want and stop them from getting lost in complex data structures? Hammond & Allinson (1987) address this last question with their travel metaphor which provides different search routines such as guide, tour, map and excursion. There are also questions about peoples memory for spatial location. According to Jones and Dumais (1986) spatial memory deteriorates more rapidly than symbolic memory as the number of items increases. It is also important to bear in mind that retrieval routes which are useful in real-world retrieval situations need not limit the choice of retrieval routes in computerized systems. If, for example, I want a specific document there is no reason to tire myself out sifting through files in an electronic office or walking up and down the shelves of an electronic library. More direct efficient access based on content is a sensible approach. This touches on the fact that metaphors should not be taken too literally. Problems associated with recognising the boundaries of a metaphor as well as their potential to restrict the design and use of a system are related to this point and are discussed in more detail later. Robert Riekert, cited by Carroll, Mack & Kellog (1988), uses a television set metaphor to combat his pupils fears about computers in adult computer literacy classes. Interestingly this metaphor is directed more at transfering the students attitudes than their knowledge. Although people generally do not know much about the internal functioning of a television they are not afraid of it. “”"”"”"”"”"”"”"”"”"”"”"”" It is evident that the use of system metaphors is a widespread design technique, but is their any empirical evidence pointing to the way in which they may affect the user? Johnson-Laird (1981;1983) shows that our ability to solve a problem depends on the domain within which it is presented, rather than its logical structure. This is evidence that metphor manipulation can influence a persons understanding of a system. This is similar to the argument that different metaphors highlight aspects of a system and make others more difficult to see. For example when trying to understand the basic principles of an electronic circuit, people understand different aspects depending on whether they use the metaphor of flowing water or of crowds of people moving around the circuit. In Rumelhart and Norman’s (1981) three models experiment to teach people about text processors. They used a secretary metaphor to show how commands can be interspersed with text input, a card file metaphor to explain the deletion of a single numbered line from a file, and a tape recorder metaphor described the need for explicit terminators in files. The metaphors had a measurable effect on the learnability of the text processor. Foss, Rossen & Smith (1982) studied the task of learning to use a text editor by novice users. Subjects presented with an advance organiser prior to the systems manual performed slightly better than those who were not. The organiser subjects required less time to complete performance tasks and used fewer commands. The advance organiser referred to a large extent to the idea of file folders which are stored and retrieved from filing cabinets; as well as explaining how the computer follows commands and can serve as a tool. The examples in the paragraphs above demonstrate that metaphors have a measurable effect on learning, but they do not offer a definition of metaphor. How are we to recognise and know when metaphors have been created without a principled description of the mappings between the source and target domains of the metaphor? Furthermore, how can we determine how the metaphor operates in the mind in real circumstances, that is, determine what kind of constraints controll the mapping? It would be useful to be able to provide a set of guidelines for, or even predict good metaphors, rather than simply being able to diagnose them ad post facto. “”"”"”"”"”"”"”"”"”"”"”"”" In an attempt to fulfil this need three main theories of metaphor within the context of system and interface design have beed developed. Hammond & Allinson (1987) base their theory on Touangeau & Sternberg’s (1982) interactionist approach which recognises both matches and mismatches between the two domains as contributing towards the metaphor. Metaphors are measured along two dimensions, scope and level of description. Scope refers to the number of concepts the metaphor addresses. Command-driven systems, for example, typically have a large number of metaphors of narrow scope; whereas the Unix hydraulic metaphor, which makes use of concepts such as pipe, is less restricted. The desktop metaphor is classed as having a wide scope and attempting to capture as much of the task it supports as possible. The level of description refers to the type of information a metaphor is expected to communicate. Hammond & Allinson appear principally concerned with task domain metaphors which aid the user’s understanding of tasks supported by the system. In the simple applications model which they propose they distinguish four levels: task, semantic, lexical and physical. As an example they give the metaphor of a tour around a given topic in an on line encyclopaedia. They suggest that the physical level, for example a commantry given by a guide, will be rejected, but other levels will be useful: At the task level, the user wants to find out about a topic that could plausibly be the subject of a tour; at the semantic level, the notion of a predetermined sequence of places maps well onto known or plausible system facilitates for viewing a series of displays; at the lexical level, the terminology of tours may relate to words or icons that the user may have noticed…people do not expect metaphors to explain everything. Nor may the absence of non-central features of the metaphor – such as a guide’s commentary – be important (H&A 1987, p80) Halasz & Moran (1982) distinguish conceptual models, analogies and metaphors. Conceptual models are intended as highly accurate, complete descriptions of a domain in some abstract format such as a graph or flow chart. Analogies involve structure mapping; at some level of abstraction the unknown system behaves the same way as the known system. Only part of an unknown system is similar to the known system in the case of metaphor, that is, only the most salient points of comparison. Halasz and Moran believe that analogies can cause faulty reasoning about a system and limit the view a user has of a system. They argue that conceptual models should be prefered for learning and suggest metaphors which explain just one or two aspects of a system (i.e. of narrow scope) may be useful. In contrast to this structural approach, which sees mismatches as potentially confusing, Carroll & Thomas (1982) and Carroll & Mack (1985) present what they call an active learning theory of metaphor, which involves a incremental process of association and consolidation of knowledge, guided and constrained by the user’s current task and goals. They see the indeterminate, incomplete nature of metaphor as promoting active and creative thought. The mind of the learner: takes an open-ended, incomplete – even indeterminate – kernel comparison, and makes it a focus for the self- initiated construction of new knowledge. (C&M 1985, p?) In the same vein they do not see mismatches to be necessarily problematic: Salient dissimilarities – in the context of salient similarities – stimulate thought and enhance the efficacy of the metaphor as a learning vehicle. (C&M 1985, p47) They thereby challenge Halasz & Moran’s opinion that conceptual models are good for supporting learning, saying that this embodies a passive learning theory where the learner has nothing left to do and consequently does not learn. The scope of this essay does not allow an indepth discussion of each theory, however based on the three theories there follows a discussion of important issues concerning the use of metaphors for users and designers. One highly relevant question is how to fit the metaphor to the user. Different user populations need different metaphors depending on their background knowledge. For example, CoSy, a conferencing system for distance learning, uses a system concept of the electronic campus shown in figure 1. The users of the system can roughly be divided into two groups: the academic staff and the students. The concept is seemingly intended to demonstrate how the system can be used and to shape the users’ attitude towards the system by use of a familiar concept in university education – the campus. This is perhaps a misconception: the campus will be familiar to the academic staff, but it may not be to the students, since they are, after all, involved in distance learning. Users also need different metaphors according to the amount of experience they have with computer systems. Novices with no prior experience will need metaphors based on non-computer domains; whereas those who do have prior experience can make use of metaphors which draw on this experience. For example, when creating a new version of a word processor the designer should obviously not present a whole set of new metaphors, but rather draw on and perhaps elaborate the existing ones. Considerations such as these can discipline and direct the designer and prevent the use of metaphors which although useful for the designer may make the system opaic to the user. The metaphor itself can act as a focus for design activities; it can be used to help define and structure facilities subsumed and excluded from it, as well as suggesting additional useful facilities otherwise overlooked. A further complex issue is how mismatches influence the use of a metaphor. Mismatches between the metaphor and the target domain can occur in either direction. Sometimes salient aspects of the metaphor do not map to the target domain, as in the typewriter where the majority of word processor keys behave like their typewriter equivalents except for the space bar. Halasz & Moran (1982) argue that this is a source of confusion for the learner and that it is better to use metaphors which just demonstrate one or two aspects of a system. This begs the question of how designers should signpost the boarders of a metaphor. Carroll & Thomson (1982) suggest that the user manual should “explicitly indicate to the user how the metaphor cannot be pushed too far”. Carroll & Mack (1985), in contrast, make the point, as already mentioned, that mismatches stimulate the user to learn about the system. Carroll, Mack & Kellog (1987) talk about the necessity of ensuring that the increase in complexity which this causes should not outweigh the benefits of the metaphor. Carroll & Mack are in favour of metaphors of broader scope rather than a collection of metaphors. They do point out, however, that it is sometimes not possible to cover all aspects of a system with a single metaphor. In such cases they suggest the metaphors should be taken from the same task domain, but care should be taken that their functions do not overlap. Hammond & Allinson (1987) warn against the attempt to adhere strictly to metaphors of wide scope across several levels of description and point out that this can make a system less flexible and more cumbersome (c.f. sifting through virtual office draws in a Spatial Data Management System). Instead they propose that the system should improve upon the metaphor and not be bounded by it: a metaphor can support partial mappings of knowledge with no ambiguity provided it is well chosen and provided the system is designed appropriately around it (H&A 1987, p87) Unfortunately they do not explain exactly how to achieve this. There are often important aspects of the target domain which do not map to the metaphor and are thus hidden. A metaphor of this type can cause a designer to fail to see important aspects of the task a system should support. This was a lethal error made by the designers of Cognoter, a “multi-user idea organising tool”, which led to “serious breakdowns in the system”(p54). Cognoter was used by people in the same room who could consequently see each other, but the textual component was designed around a “parcel post” model of communication. items are packaged and sent by the speaker, and then unpackaged and decoded by the reciever…if the reciever does not open his “mail” right away, he may end up with a bunch of stuff with no particular order. (p64) This metaphor entails that the “items” of communication are independent of people and temporal contexts. This masks other aspects of the communicative process and it was a failure to support these aspects which lead to communication problems. An interactive model of communication was subsequently adopted. This same masking effect can be a problem for users; the same metaphor which was initially useful in understanding the system can perhaps later become a barrier to further learning and prevent users exploiting the system to the full. The example of the electronic whiteboard given earlier is a case in point. It may not be obvious that the ideas saved on the board can be manipulated and reorganised, which is not the case with the real-world equivalent. Marshall, Nelson & Gardiner (1987b) also make this point: …by typing an interface to concepts which prevail in non-electronic environments, one is not taking full advantage of the benefits that can accrue from using the electronic medium…For example, a filing cabinet can be as restrictive as the real-life filing cabinet How can problems of this kind be overcome? The designer can highlight the provisional nature of metaphors and point out deviations from the real system at the time of initial learning and make the limits explicit. The designer should also keep in mind from the beginning that metaphors are presented to the user for an overview and may loose their usefulness with time. He or she could try to make them fit in with how a user eventually comes to conceptualize the system or task. The literature which exists on the subject of metaphor for users and designers of systems demonstrates that there do exist researches in the field of HCI who take metaphors seriously and current interest in virtual reality seems in turn to be stimulating interest in this area. However significant questions do remain. For example, can we predict a “good” metaphor? The answer is probably not. Hammond & Alinson (1987) develop an approximate model and Carroll, Mack & Kellog (1987) outline a structured methodology for designers to choose metaphors, although they admit that “there is a role for sheer invention” (C,M&K 1987, p78). An obvious question is whether metaphors are not perhaps the same thing as what most people would refer to as models (system models and user models). Perhaps in some respects they are. It is recognised that that models by their abstract nature are not complete and may require some degree of interpretation. However metaphors are not just incomplete; they are by definition partial/open-ended and even indeterminate. Models are often taken quite literally. An important point to remember about metaphors is that they are not intended to be taken literally. Margret Thatcher is not literally made of iron. In this respect metaphor possibly throws light on the way to view system concepts, that is, the concept of metaphor itself highlights the issues designers must consider, which may not otherwise be so obvious. The question is whether there is anything that it hides. Bibliography Carroll, J. M. & Mack, R. L. (1985). Metaphor, computing systems, and active learning, in International Journal of Man-Machine Studies, 22, 39-57 Carroll, J. M. & Thomas, J. C. (1982). Metaphor and the cognitive representation of computing systems, in IEEE Transactions on Systems, Man and Cybernetics, 12, 107-116 Carroll, J. M., Mack, R. L. & Kellog, W. A. (1988). Interface metaphors and user interface design, in Handbook of Human-Computer Interaction, M. Helander (ed), Elsevier Science Publishers, North Holland Gentner, D. (1980). The structure of analogical models in science. BBN Technical Report 4451 Halasz, F. & Moran, T. P. (1982). Analogy considered harmful, in Proceedings of the Conference on Human Factors in Computing Systems. Gaithersburg, Maryland, pp383-386 Hammond, N. V. & Allinson, L. J. (1987). The travel metaphor as design principle and training aid for navigating around complex systems, in People and Computers III, Diaper, D. & Winder, R. (eds) Jones, W. P. & Dumais, S. (1986). The spatial metaphor for user interfaces: Experimental test of reference by location versus naming. ACM Transactions on Office Information Systems, 4, 42-63 Mason, R. (1989). An evaluation of CoSy on an Open University course, in Mason, R. & Kaye, A. (eds). Mindweave: Communication, Computers and Distance Education. Oxford, Pergamon Press