Metaphor-Based Interactions

Marko Djordjevic

ABSTRACT

The use of metaphors in the design of human-computer interaction (HCI) has been increasing as the graphic user interfaces have become popular in recent years. Metaphors are the tools used to link highly technical and complex software with users' bodily experience and other everyday interactions with the external world. Therefore, the efficiency of an HCI and the usability of a computer system may largely determined by the appropriateness of the metaphors used. This report summarises research publication by different authors. These publications analyzed a number of existing HCI metaphors and developed guidelines for determining basic metaphors that should be used in the design of HCI to maximize the intelligibility and minimize the intended users' mental workload. In conclusion, for designers good metaphors are necessarily to look after both sides of functionality and visualization (aesthetics and consistency, etc.) of the user interface (UI). For users good metaphors would be able to help them get acquainted with the new interface and direct them into acting a certain way while reducing the risks of mistakes. The metaphors should be an intuitive access to all functionality.

KEYWORDS

Metaphor, interactions, human-computer interaction, interface

INTRODUCTION

Metaphors have been shown to help designers generate creative design decisions, maintain consistency in the interface, keep the number of design decisions manageable, and provide rationale for the design decisions adopted. Metaphors directly and indirectly influence the design of UI [6]. What this means is that interface design could be created by looking at a metaphorical influence since most of the designers use this form of thinking (without them even knowing it) to manage their interfaces, or it could be that they decide on design activities by directly looking at the supporting metaphor.

Gillan and Bias outlines two specific uses of metaphors in aiding designers. Metaphors increase consistency and commonality in UIs, while also reducing the number of design decisions. Tepper says that metaphors encourage creative thinking and provide a valuable communication tool in design process [6].

Carroll et al. classified three general approaches to metaphor in human-computer interaction (HCI): The *operational* approach describes metaphor as an educational resource for the user. The *structural* approach evaluates specific relationships between source and target domains of the comparison. The *pragmatic* approach anticipates breakdowns and mismatches between different user goals and the presentational metaphor, regarding the mismatches as an opportunity for richer experience [2].

Metaphors are a great way to get people to spend time and learn to use the most difficult of interface designs. With enough exposure, these odd connections (user's mental model) start to feel like they are obvious and intuitive [5]. Some examples of more successful metaphors include desktop, windows, menu, widget, etc.

DEFINING "METAPHOR"

It has been argued that our conceptual system, the term in which we think and act, is essentially metaphorical in nature [2, 5]. Metaphors are the tools used to link highly technical and complex software with users' bodily experience and other everyday interactions with the external world. As described in publication by Chuang M.C. and Lo I. a good metaphor is necessary to oversee both sides of functionality and the visualisation of the UI [4].

Most of the publications referenced have described metaphor as a central principle of UI design. Metaphor is a fascinating case study of such importance because it has been in many ways a central theme of the expansion of human computer interaction (HCI) as a design research discipline. Weinschenk conveyed that "*metaphors are the tools we use to link highly technical, complex software with the user's everyday world*". Metaphors make it easy to learn about unfamiliar objects [2].

When metaphors are considered in HCI, a computer UI might be considered to be a kind of *literary* description. A sort of representation made to help users understand abstract operations and compatibilities of the computer. These abstract capabilities are therefore presented as thought they were something else that the user might already understand [2].

METAPHOR MISMATCHES

Understanding metaphor mismatches is a critical part to developing effective UI design strategy because they provide much of the strength of metaphors. It is one of the greatest challenges in getting metaphor design right. Similarities and dissimilarities between source and target domains both play a role in how the metaphor is going to be interpreted by the user. Mismatches help users determine the target domain by emphasizing differences between the two domains; therefore they can enhance curiosity toward the system. However, mismatches also bring their own set of problems if they are overlooked by the users, creating assumptions about the system functionality and features that are not valid [6].

Metaphor mismatches can occur due to computer functionality that cannot be directly linked to the real world, because the extended functionality in computer systems is more flexible than the concrete reality. The problem then arises how to represent computer functionality that does not exist in the physical world with metaphor taken from real world. That is why designers tend to use a set number of composite metaphors with the features that are literal and *magic* [6]. Literal features are those that are consistent with the metaphor and have real-world analogies. Mandel said *"Real world metaphors allow users to transfer knowledge about how things should look and work"* [2]. *Magical* features are those that extend beyond the metaphor and generally do not have structure found in the world.

Metaphors do not need to cover every aspect of the computer functionality, nor do *magical* features necessarily undermine real world metaphors. Adding *magic* features and functions that violate the basic metaphor is appropriate as long as they do not mislead the user. Over time, these *magical* features and functions will become the literal metaphors of tomorrow [5, 6].

THEORETICAL MODEL OF METAPHOR

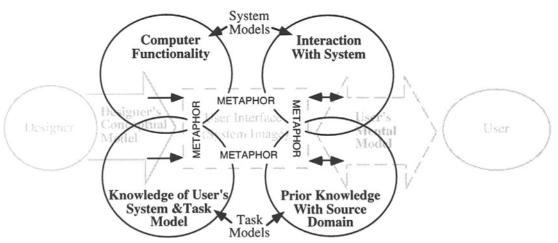

Figure 1 Model of the user interface (Adapted from Norman) [6]

Figure 1 shows in the watermark the framework of how designers and users interact with the interface and their mental representation. Designer's conceptual model is a model that the designer has formulated to accommodate full functionality of the system. It includes tasks, experiences; capabilities and limitations users would be faced when using this system. User's mental model shows how users interact with the system, bringing their past experiences to go about the system and changing their knowledge about the system's features and functionality as they interact. Ideally over time the user's mental model will develop to designer's conceptual model [6].

Both the designer's and user's model include a model of the system model (computer system) and a model of the task model (task domain). The designer's system model is an abstract view on how the system should operate.

The metaphor is implemented as a tool through the system to help users create a model that would be compatible with the designer's conceptual model. The metaphor does this by linking the models together as illustrated in Figure 1. It is important that designers attempt to predict how user's mental models will change over time as a result of interaction with the system or changes in the knowledge of the system's features or functionality [6].

A clear distinction should be made between user's mental model and designer's conceptual model. A user's mental model is often a simplified designer's conceptual model. There have been some debates on whether it is a smart idea to have a user's mental model representing the designer's conceptual model.

METAPHOR CRITIQUES

Metaphors are often critiqued for not having the ability to fully represent the complexity of the system's functionalities. Some authors suggest a more abstract structure of the system would be more appropriate [6]. Recent textbooks warn against reliance on metaphors, and criticize the designs that were heavily influenced by metaphor through the 1980's and 1990's. One of those designs included General Magic's Magic Cap in 1994 and Microsoft's Bob in 1995. Importance of metaphors was so convincing in those times that the developers were stunned to find out that extra-realistic pictorial metaphors did not succeeded to the same extent that the relatively abstract "desktop" and "windows" metaphors did [2].

Nelson critiqued "what I object to is severalfold: first, these mnemonic gimmicks are not very useful for presenting the ideas in the first place; second, their resemblance to any real objects in the world is so tenuous that it gets in the way more than it helps; and third . . . the metaphor becomes a dead weight . . . The visualizations become locked to some sort of continuing relation to the mnemonic. It becomes like a lie or a large government project: more and more things have to be added to it" [2].

After the disappointment from designing these overly-metaphorical UIs, textbook authors are now being more cautions. More recent textbooks who mention metaphors tend to warn that they should be treated with care. Some authors even campaign against them, for example Cooper writes: "Searching for that magic metaphor is one of the biggest mistakes you can make in UI design" [2].

But these abstract structures provide less flexibility and lack power and expression. Metaphors provide a medium that encourages learn ability and motivates people into continuously using the computer. Learning is viewed as an active process where computer users prefer to explore a new system, rather than learn by more traditional methods of following instructions [6].

The use of metaphor retains creative potential that may lead to users' wild and unruly interpretations escaping the control of the designer's intent [2]. Therefore it is important for the designers to predict the way users might interpret the metaphor and how the mental models would change over time as a result of interacting with the system, and how these interactions relate to their prior knowledge. Faulkner said "*designers of systems should, where possible, use metaphors that the user will be familiar with*" [2].

METHODOLOGY FOR DEVELOPING USER INTERFACES METAPHORS

Structured methodology for developing UI metaphors consist of five stages [6]:

i. <u>Identify system functionality:</u>
Before any metaphors can be used to model a system, it is required to know what the system functionalities are that need to be mapped. That means there needs to be requirements analysis and functional specification which would identify functions and features which users of the system would need to be able to reach their goals.

ii. <u>Generate Possible Metaphors:</u>
Generating metaphors is mainly a heuristic process, one that could be really creative and difficult to structure. This could be done by looking at previous systems and trying to design and apply similar metaphors for the new system. Designers also try to generate ideas by sketching different metaphors. This visual method to developing metaphors offers a more tangible approach to visualizing possible alternatives. Also market feedback from customers may lead to a fresh perspective. Smyth et al. claimed that in the customer attempt to describe the functionality of the system, they are forced to employ rich metaphors from the source domain.

iii. <u>Identify Metaphor-Interface Matches:</u>
Identify metaphor matches form the basis of the primary relationship between the source and target domains. This is done through goal-oriented user scenarios.

iv. <u>Identify Metaphor-Interface Mismatches:</u>
Mismatches are inevitable, as talked about in the previous sections. It is crucial to determine when they will lead to erroneous actions, rather than leading to new insights. Therefore they have a great role in deciding if the metaphor is appropriate for use in that particular interface. As with metaphor matches, best way to identify metaphor mismatches is through goal-oriented user scenarios.

v. <u>Manage Metaphor-interface Mismatches:</u>
Composite metaphors could be used to manage metaphor mismatches, by creating a match with one metaphor for the mismatch of another. Also composite metaphors offer an alternative view of the system, giving the user a greater insight to what the mismatch implies. When trying to account for mismatches, it is important to keep in mind that the focus of the design is to communicate the functionality of the system, rather than trying to mimic every aspect of the source domain.

THE MODEL OF ANDERSON ET AL.

The paper by J.L Alty and R.P. [1] Knott uses the model of Anderson *et al.* to investigate the concept of "conceptual baggage". Conceptual baggage to Anderson is the proportion of those features of metaphors that do not match the system functionality compared to those that do. Anderson has shown evidence that conceptual baggage did play an important role of the overall effectiveness of the metaphors in the interface. This will be discussed in later sections of the report. This model helps designers investigate metaphoric mapping issues in the way that it leads them into discovering the wrong conclusions about the designers' conceptual model. By doing this, designers could think about how user's mental model may not map well initially with the designers conceptual model [1, 6].

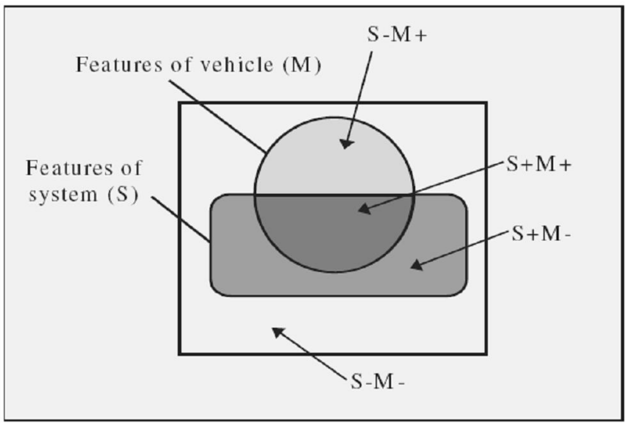

Figure 2 The Anderson et al. pragmatic model [1]

The four areas (in what might be considered a Venn diagram) are:

S+M+: features in the system supported by the Metaphor,

S+M-: features in the system not supported by the Metaphor,

S-M+: features implied by the Metaphor but not provided by the system,

S-M-: features not implied by the Metaphor nor supported by the system.

Findings of the Anderson

The Anderson's model [1] was used on the office-based prototype system which integrates digital broadband telecommunications infrastructure. The prototype system was designed to broadcast the availability state of each individual user using the system. Each user of the system was represented as a graphical icon which was seen by all other users of the system. The representation of the icons would show the availability state of a user and would initiate a communication between users. Initially the prototype's design suggested that each individual user's availability can be generally be allocated to one of three states:

i. Available for communication.
ii. Busy but interruptible.
iii. Not available for communication.

In order to describe the relationships between the system and vehicle (graphical icon) for each of the three pairings, it was necessary to look at the features of each individual vehicle in respect to the proposed system functionality. Therefore before any experimenting was performed, Anderson et al. used techniques suggested by Carroll *et al* [6] to consider the mappings between the vehicle and the system by looking at the levels of tasks, methods and appearances in representative set of scenarios. The results of this analysis help in grouping the attributes of the vehicle-system paring to one of the four categories in Anderson's model [1].

Office Doors Metaphor

The first vehicle-system pairing [1] used *office door* as a vehicle for representing availability of a user. It uses images of an open door to correspond to user being 'available to communicate', a partially opened door to signify 'busy but interruptible' and finally a closed door to signify that the user was 'not available for communication'. The characterization of the relationships between this vehicle and the system is shown in Figure 3.

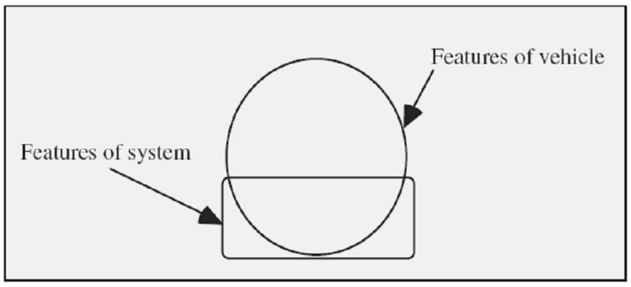

Figure 3 Characterization of office doors and system pairing [1]

Observing this model we see that the operators thought that the *office door* metaphor was considered an appropriate vehicle, since most of the system functionality was accounted for by it. This is due to the assumption that the user mental model would be able to easily adopt the metaphor since it suits this particular context. From this characterization certain predictions can be made about the expected pattern of use by the users of this system. Firstly it could be expected that subjects undertaking this experiment would find it very easy to use even if they have not encountered it before, not only because the metaphor seems contextually relevant, but also because the amount of system features not covered by the metaphor is fairly low. For this reason it is expected for the subjects to explore the system and successfully utilize the underlying functionality. However, it is predicted that because there is lot of conceptual baggage present in this metaphor, it is expected for the subjects to eventually get frustrated by the lack of functionality they might expect from the system. For example individuals might see that a closed door might be interruptible, and they would be frustrated to found out that the system does not respond [1].

Dog Metaphor

The second vehicle-system pairing [1] adopted the *dog* as a vehicle for representing the availability of a user. They used an attentive dog to represent 'available for communication', a digging dog to correspond with 'busy but interruptible', and finally a sleeping dog for 'not available for communication'. The characteristics of the relationships could be seen in Figure 4.

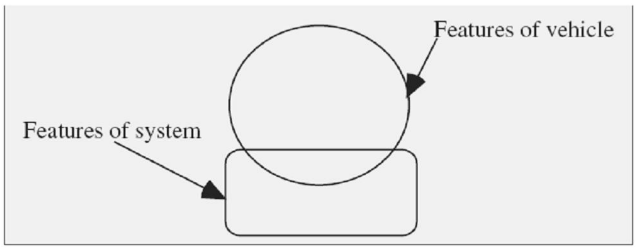

Figure 4 Characterization of dogs and system pairing [1]

In this case, unlike in the previous one, it is expected for the users to potentially miss out on most of the system underlying functionality due to the lack of contextual relevance. Such a characterization would lead to different perceptions to what the system is designed to do, not only because the metaphor is less intuitive, but also the fact that most of the system functionality would be unexplored. This vehicle also suffers from lot of conceptual baggage. For example subjects may perceive that each individual dog could be trained to allow communication for specified people [1].

Traffic Lights Metaphor

The third and last vehicle-system paring [1] in this experiment used *traffic lights* as a vehicle for representing availability of the users. Green light corresponded to ‘available for communication’, an amber light for ‘busy but interruptible’ and red light for ‘not available for communication’. The characteristics of the relationships could be seen in Figure 5.

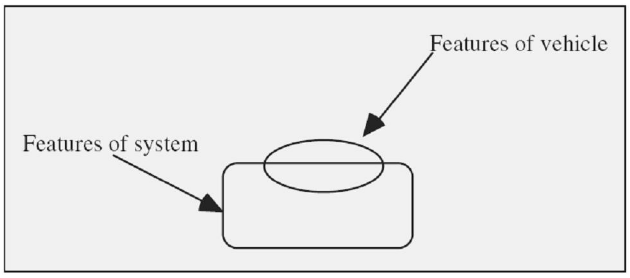

Figure 5 Characterization of traffic lights and system pairing []

In this pairing it can be observed that there are few potential relevant features of the vehicle that are not supported by the system. In this instance, there is considerably less conceptual baggage than in the previous two situations. Similarly to the dog scenario, there is very little system functionality accounted for by the features of the vehicle because of the lack of contextual relevance. It is predicted that the subjects would not initially find the system intuitive. For the same reason it would be expected even if subjects do explore the system and become familiar with the functionality, the boundary between system features that are not covered by the metaphor to does that are would become apparent. Finally, because of the lack of conceptual baggage it would be expected that the subjects will be greatly able to distinguish those features of the vehicle that do play a part of the system to those that do not [1].

Experiment Findings

An experiment was then designed and carried out to investigate the viability of the model by utilizing the interface metaphors office doors, dogs and traffic lights. Experimental data was collected using a combination of verbal protocol, activity capture using video and questionnaire techniques.

It was clear from the results that the intuitive nature of the *office door* interface metaphor caused the subjects to make incorrect assumptions concerning the nature of the underlying system functionality. This meant that the subjects were not able to distinguish what functionality was part of the underlying system to what was not. This is mainly due to do effects of the conceptual baggage that this vehicle has.

In the case of *dogs* interface metaphor, subjects were better able to identify system functionality that was not supported by the vehicle, than functionality that the vehicle suggested but was not present in the system. Most of subjects required a verbal explanation of the representation of system states at the start of the task. This is due to the lack of contextual relevance in this interface metaphor. Therefore, whilst a degree of conceptual baggage could be expected, due to the verbal explanations this effect was reduced.

In the case of *traffic lights*, subjects were better able to identify system functionality that was supported by the vehicle than functionality that was suggested by the vehicle but was not present in the system. Majority of subjects expressed a need for a verbal explanation since they found the vehicle non-intuitive. Once the subjects became aware of the mapping between vehicle and system, actual understanding of the interaction was superior to that in either of the other two vehicle-system mappings.

As shown in these finding, Andersons model could be used to provide designers a mechanism for choosing among possible metaphors before system is built. This model is complimentary to the active learning theory (as described in a previous section) and provides some tools for evaluating and selecting UI metaphors [1].

CONCLUSION

In conclusion, for designers good metaphors are necessarily to look after both sides of functionality and visualization (aesthetics and consistency, etc.) of the UI. For users good metaphors would be able to help them get acquainted with the new interface and direct them into acting a certain way while reducing the risks of mistakes. The metaphors should be an intuitive access to all functionality. That means the user should be not only familiar with the metaphor domain but also able to perform the mapping between source and target domain [3].

New techniques need to be developed that obtain valid knowledge representations (user's mental models) and the means for transforming those models into design specifications that are specifically useful for metaphor development and elaboration [6]. Search for improved interaction metaphors is an active research and development area. As with other types of semantic change in human language and cultures, what may at first have been marked as strange may become common. Over time these metaphors might become definitional identities, rather than conceptual mappings [3].

FUTURE WORK

Unfortunately, most of the HCI metaphor theories are based on a single group of researches' observations and data, and they lack substantial empirical support. Much additional theoretical and empirical work is needed from the HCI community [1, 6].

In lazy moments metaphors become the *thing*. They become reified and reused, and some designers might be confused with fact or rules for design. Scaffolding reasoning through metaphors can lead to problems of initial understanding, design rigidity, and overextension. Perhaps most interestingly for the global world of internet based interaction and communications problems of translation and derivation could occur [5]. Problems of cross-culture understanding that depend on the ubiquity of metaphor in human language will be faced by software agents in going beyond lexical surface meaning [3].

REFERENCES

1. Alty, J.L., & Knott, R.P. (1999) Metaphor and Human-Computer Interaction: A Model-Based approach. Lecture Notes in Computer Science, pp.1-15. Retrieved on April 18th, 2008, from http://www.springerlink.com.ezproxy.auckland.ac.nz/content/kltbh3hdc5jugjvr/fulltext.pdf.
2. Blackwell, A. F. (2006) The Reification of Metaphor as a Design Tool. ACM Transactions on Computer-Human Interaction 13 (4), pp.490-530. Retrieved on March 24th, 2008, from http://portal.acm.org.ezproxy.auckland.ac.nz/citation.cfm?doid=1188816.1188820.
3. Chrystopher, L. N. (1999) Computation for metaphors, analogy, and agents, pp.1-11. Retrieved on March 24th, 2008, from Voyager Database.
4. Chuang, M.C., & Lo, I. (2007) The Usability of Metaphors with Different Degree of Abstract in Interface Design. Lecture Notes in Computer Science, pp. 343-352. Retrieve on 25th March, 2008, from http://www.springerlink.com.ezproxy.auckland.ac.nz/content/g747283h87814666.
5. Churchill, E. (2008) What's In A Name? Idioms, Metaphors, and Design. Interactions 15 (1), pp. 6-10. Retrieved on March 25th, 2008, from http://www.scopus.com.ezproxy.auckland.ac.nz/scopus/redirect/linking.url?targetURL=http%3a%2f%2fdx.doi.org%2f10.1145%2f1330526.1330530&locationID=2&categoryID=4&eid=2-s2.0-38649095593&issn=10725520&linkType=ViewAtPublisher&year=2008&origin=resultslist&dig=890f8d96187fee9af8b3f64dce517f8c
6. Helander, M., & Landauer, T.K., & Prabhu, P.(1997) Handbook of Human-Computer Interaction (2), pp.441-457. Retrieved on April 19th, 2008, from http://www.sciencedirect.com.ezproxy.auckland.ac.nz/science/book/9780444818621.