A Taxonomy of User-Interface Metaphors

Pippin Barr, Robert Biddle & James Noble
School of Mathematical and Computing Sciences
Victoria University of Wellington
Wellington, New Zealand
{chikken,robert,kjx}@mcs.vuw.ac.nz

ABSTRACT
Although metaphor is a commonly used device in the design of user-interfaces, it is not rigorously understood, and most guidance stops at the recommendation of its use. In this paper we seek to provide a systematic taxonomy of user-interface metaphors, based on and extending the framework of Lakoff and Johnson. We then suggest that some usability heuristics emerge directly from analysis of the taxonomy. We conclude that the taxonomy and heuristics may provide appreciable benefits in user-interface design and evaluation, and address some of the criticisms of metaphor use that have been made.

Keywords
metaphor, user-interface, heuristics

INTRODUCTION
The idea of using metaphors to help leverage existing user knowledge in user-interfaces has been around for as long as graphical user-interfaces have existed, and indeed longer. Additionally, metaphor use is recommended by most interface-design handbooks (see, for example, [10]). Recently, discussion has become more circumspect and taken new directions. In particular, there has been concern over the places where a metaphor does not apply [9, p.131], and also whether metaphor really achieves the benefits usually claimed, such as increased learning rate and so forth [2].

Despite criticism, metaphor is still used frequently by user-interface developers. What is more, metaphor seems to be used with little understanding of its underlying structure, and is often used unknowingly. The general understanding is that a user-interface metaphor "is the process of representing the computer system with objects and events from a noncomputer domain" [14, p.273]. In this way a desktop is used to represent the file-system of most of today's personal computers. Despite the simplicity of the definition, there are very few rigorous approaches to the use and understanding of user-interface metaphors (although see [3] and [5] for examples of advice). In this paper we seek to analyse the meaning of metaphor in user-interface design, and to better see when metaphor is being used.

We start by offering a systematic taxonomy of user-interface metaphors based on Lakoff and Johnson's work on metaphors as a key cognitive device [8]. We then look at metaphoric entailments as a way of characterising metaphor structure. Finally, we note briefly that some of the issues made apparent by the taxonomy suggest some useful heuristics for the design and evaluation of user-interface metaphors. A more detailed examination of the practical aspects of this taxonomy and the heuristics is left to future work as the scope of this paper is limited to examining how Lakoff and Johnson's framework can be applied to user-interfaces.

A TAXONOMY OF INTERFACE METAPHORS
We decided to ground the taxonomy in the work of Lakoff and Johnson as it provides a strong philosophical and linguistic basis from which to work. Given that metaphor is originally a literary and linguistic concept, this seems a sound approach to take. By applying the framework described by Lakoff and Johnson [8] to user-interface metaphors we can enhance our understanding of them, as well as provide a vocabulary for their discussion. Clearly, using Lakoff and Johnson's model is not the only way to approach a taxonomy of interface metaphor, but it is at least a good start.
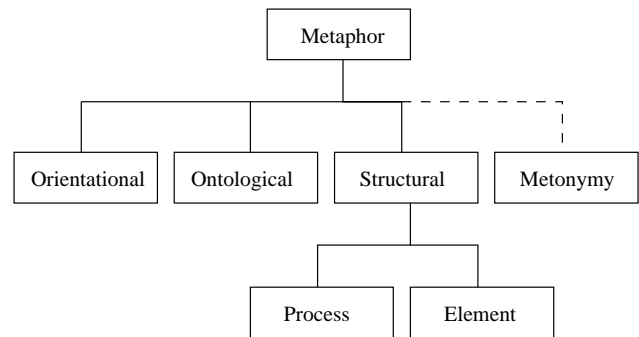

Figure 1: Diagram of our taxonomy of metaphor

It is important to note that the categories provided in this taxonomy are not all mutually exclusive. There is a hierarchy involved which is depicted in figure 1. The first several categories correspond exactly to categories in the framework of Lakoff and Johnson, but as applied to user-interface metaphors. The final two categories, process and element metaphors, are extensions specifically developed by us to describe user-interface metaphors. In each case we will explain the grounding in Lakoff and Johnson, and then go on to show how the categories can be used to characterise particular kinds of user-interface metaphor, often with examples.

Orientational Metaphors
Lakoff and Johnson characterise orientational metaphors as metaphors which "give a concept a spatial orientation" [8,

p.14]. Very simply, then, an orientational metaphor involves *explaining a concept in terms of space*. An example of this is the metaphor HAPPINESS IS UP[1] as evidenced in phrases like "I'm feeling *up* today" and "he has such *high* spirits". In this case we can use the spatial orientation of "up" to talk about happiness.

An important aspect of orientational metaphors is that they organise "a whole system of concepts with respect to one another." [8, p.14] This means that all the concepts that use "up" as a spatial metaphor, for example, tend to be related to each other. Thus, we find that GOOD IS UP and IMPORTANT IS UP, and because of this there are certain associations between those concepts. Orientational metaphors do not simply structure our thinking about one concept, but about systems of concepts.

Another key point to note is that orientational metaphors are strongly based in our physical and cultural experiences of the world. Because of this they might often not appear to be metaphoric at all, but simply the way things are.

Because orientational metaphors are not generally obvious, it might be difficult to see their relevance to interfaces. In fact, the main point of discussing orientational metaphors here is that they are *already* used often in user-interfaces. In particular, orientational metaphors tend to be used for *quantification* and *navigation*.

An example of quantification is the UP IS MORE metaphor, which applies in general to all vertical sliders and other interface elements that involve increasing something by associating it with an upwards direction. The most obvious experiential basis of this is that "if you add more of a substance ... to a container ... the level goes up." [8, p.16]

An example of a navigation metaphor is PROGRESS IS TO THE RIGHT. When we move through a task wizard, for instance, the 'next' button often literally *points* to the right. Note that we do not actually *move* right, but progress to the next stage of the wizard. This sort of metaphor probably comes from our experience of reading text, and is thus highly cultural as not all cultures read from left to right.

Ontological Metaphors

Lakoff and Johnson claim that ontological metaphors arise when "our experience of physical objects and substances provides a further basis for understanding" [8, p.25]. Ontological metaphors, therefore, *explain concepts in terms of the very basic categories of our existence* such as objects and substances.

An example of an ontological metaphor used in everyday life is that TIME IS AN OBJECT, which we can observe when we make statements such as "I don't have enough time" or "thanks for giving me your time". Treating time in this way enables us to understand it and especially to quantify it. Lakoff and Johnson suggest that ontological metaphors serve many purposes, such as referring, quantification, identification of aspects, identification of causes, and helping to set goals and motivate actions [8, pp.26-27].

---

[1] Writing the expressions of metaphors in small-caps is a stylistic decision taken from Lakoff and Johnson [8].

While ontological metaphors might initially seem too fundamental to be of use in user-interfaces, this turns out not to be true. In fact, *because* they are basic, ontological metaphors are to be found *everywhere* in our interfaces, so it is imperative that we know they are there, and how they work.

If we consider the purposes of ontological metaphors mentioned above, we can see that these are all things we wish to do in making user-interfaces work. Quantification is plainly something we are constantly doing in user-interface design, and this leads to metaphors which present elements of the system as *objects*, things which can be quantified. Examples of this are plentiful, but consider THE FILE IS AN OBJECT. This enables us to talk about the *size* of files, and also to give them *locations*, and various other useful characteristics. To give another example, we frequently use ontological metaphors to identify causes in the system. Thus we might tell the user that "an error prevented the system from opening that file", thus using the metaphor ERRORS ARE ENTITIES (capable of causing and preventing things).

Structural Metaphors

Structural metaphor involves characterising the structure of one concept by comparing it to the structure of some other concept. Following the terminology of semiotics defined by Saussure we will use the terms "signifier" and "signified". Thus, in the desktop metaphor, the desktop is the signifier and the file-system is the signified [4]. These concepts can be abstract, actual objects, events and so forth. Effectively we obtain structural metaphors when we make ontological metaphors more specific, thus we move from X IS AN OBJECT to specifying the object that X is.

The key here is that the signifier is able to reveal interesting and important properties of the signified through the process of metaphor. The classic example is the ARGUMENT IS WAR metaphor, which explains many things about our experiences of arguments. In particular, it is a large reason why we "feel embattled" and "attacked" and so forth. From this we see that metaphors are not merely linguistic but relate directly to our *experiences* of the concepts.

Structural metaphors differ from orientational and ontological metaphors in that they deal more directly with our experience of everyday life. They are used to compare our concepts with objects in the world such as cars, artworks and desktops. In this way, they are the closest to our conscious minds, while orientational and ontological metaphors are more subconsciously apprehended.

Structure is a crucial factor when dealing with user-interfaces. Specifically, the basic task of a user-interface is to help the user in coming to grips with the correct *system image* [13]. By using metaphors based on already understood concepts and objects to explain the system's structure we are utilising a key human cognitive skill. The benefit is that, if the user has never encountered the system before, they have no clear path to discovering how it works; but when we indicate the system structure by metaphoric means, the structure becomes more immediately apparent.

The way we can examine structural metaphors is by looking at their *metaphoric entailments*. These will be discussed

later in this paper, but the essential point is that in using a metaphor we are implying that the signified has certain properties and functions. Metaphoric entailments are a way of making this explicit.

By way of example, consider the metaphor USING THE DATA-STORAGE SYSTEM IS FILING. For a user with no understanding of how data-storage works, this metaphor immediately offers structure to the process because of its metaphoric entailments. Examples of these are "you can use folders to categorise your files", "you can store files" and "you can name folders to indicate their contents". All of these entailments aid the user in not only understanding the data-storage system, but also in doing productive work.

Novel and Conventional Metaphors
A key distinction raised by Lakoff and Johnson is establishing which metaphors are novel and which are conventional. Clearly this categorisation is dependent on how the group of relevant people is defined. Some metaphors which are conventional to one group may be novel to another. We need to understand which metaphors the prospective users will take as being conventional, and which they will see as being novel.

Conventional metaphors are those which are *already used by the target audience without thinking*. The main feature of conventional metaphor is that there is an established protocol for its use: the structure of the metaphor has been set out and the group which uses it understands the way it works.

A novel metaphor for a group is *any metaphor which is not a conventional one*. In this case the structure of the metaphor needs to be established instead of assumed. Lakoff and Johnson note, however, that "new metaphors make sense of our experience in the same way conventional metaphors do" [8, p.139]. This means that, while novel metaphors are far more difficult to use, they are still the same *kind* of thing as conventional metaphors. It also means that they are far more flexible and can be tailored to explain things we have not yet encountered. Additionally, a good novel metaphor will eventually become a conventional metaphor as it is absorbed into a group's way of thinking.

Conventional metaphors are already implicitly used in interfaces today without necessarily being recognised. Examples include orientational metaphors such as MORE IS UP, ontological metaphors like INFORMATION IS A SUBSTANCE, as well as more complex structural metaphors like THE DATA IS A DOCUMENT. None of these metaphors are obvious to the user of a system because they were either already conventional before user-interfaces, or because they have been so widely used in interfaces they have become conventional.

Novel metaphors, on the other hand, will be consciously perceived by users. In particular, the metaphoric entailments have not been standardised and some exploration may well take place as the user attempts to find out which are active.

It is interesting to consider what the distribution between novel and conventional interface metaphors might be. Initially it seems as though most interface metaphors would be novel, simply because programs are all different to each other. With some more thought, however, it should be clear that there are many *standardised* metaphors that are consistently reused in interfaces. Examples include THE USER ENVIRONMENT IS A DESKTOP, THE TEXT-BOX IS A CONTAINER and THE FILE IS AN OBJECT. Thus, there are very many conventional interface metaphors, often used without thinking. Nevertheless, there are also novel metaphors invented in interfaces, and these create some issues which will be touched on below.

Metonymy
The final category in this taxonomy that is directly derived from Lakoff and Johnson is not, strictly speaking, a type of metaphor. It is, rather, the notion of *metonymy*. Where metaphor *explains* one concept in terms of another, metonymy is the use of "one entity to refer to another that is related to it." [8, p.35] Thus we say "the crown" when we mean the queen, using the metonymy THE CROWN FOR THE QUEEN.

As metonymy serves a more specific purpose, it is less flexible than metaphor in enhancing our understanding of the things in our lives. Nonetheless, Lakoff and Johnson observe that metonymy is systematic in the same way [8, p.37] and also aids understanding [8, p.36]. Metonymy is systematic in that we use certain metonymies quite frequently such as THE PART FOR THE WHOLE and THE PRODUCER FOR THE PRODUCT. These overall metonymies are then used over and over again in different situations. In terms of understanding, the metonymy THE FACE FOR THE PERSON (an instance of THE PART FOR THE WHOLE), is used when we show photos of a person's face yet claim to be showing 'the person'. This reveals that we identify people by their faces and how important the face is in terms of recognising a person.

When we turn to user-interfaces we can see that metonymy is very heavily represented in what we know as icons. Icons are *not* generally metaphoric devices because they do not always explain some aspect of system functionality by their presence, but *refer* to some piece of the system. Thus we have a metonymy THE ICON FOR THE FUNCTION or THE ICON FOR THE OBJECT. Examples of these would be the magnifying glass icon in a photo manipulation program, and a document icon in the file-system.

What is most interesting about these metonymic icons is that they do not refer to "real" objects, but to objects in the *metaphoric* world defined by the collection of interface metaphors. Thus they actually refer to metaphors. The magnifying glass icon refers to the magnifying glass metaphor that we use to explore the data in a photographic image; the document icon refers to the metaphoric document which we use to store our data.

Our choice of icons, therefore, has an effect on how the user perceives the thing referred to. Just as we learn that faces are important because of the FACE FOR THE PERSON metonymy, so we find that it is the "painting" aspect of a paintbrush tool that is emphasised by that icon, rather than the ability to perform very detailed, pixel by pixel operations that are not painterly at all. We should look to the traditional metonymies of everyday life to see what sort users might be most comfortable with and most easily recognise.

Process and Element Metaphors

As we have seen, Lakoff and Johnson's analysis of metaphor provides a detailed and potentially valuable taxonomy of metaphor in user-interfaces. Before moving on to discuss the notion of metaphoric entailments, we will introduce two extra categories of specific value to user-interface design. We discuss them here to demonstrate Lakoff and Johnson's framework *can* be extended quite easily to be even more applicable to user-interface design and evaluation.

Process and element metaphors are mainly forms of structural metaphor and are roughly derived from comments by Jakob Nielsen in his paper on Heuristic Evaluation [11] concerning the distinction between the function and the appearance of a system. A process metaphor is *used to explain how some aspect of system functionality works*. An element metaphor is *a perceivable aspect of the user-interface* which is designed to aid the user in understanding what process metaphors are applicable.

We use a process metaphor when we explain some matter of system functionality by specifically comparing it to a real-world process. This enables a user to take their knowledge of how to perform a real-world task or process, and apply it to the interface without specific training. In theory this should allow a user to learn and use the interface much more quickly and comfortably.

As an example of this, consider budgeting software. Here the process metaphor USE OF THE BUDGETING PROGRAM IS THE REAL-WORLD BUDGETING TASK enables a user to immediately begin working on their budget in the same *way* as they would in normal life. Another example is the metaphor USE OF THE DATA-STORAGE SYSTEM IS FILING. It enables users to work with the file-system by using a similar process used in their real life, thus transferring their knowledge of filing to the computer.

An element metaphor is chiefly used to cue the user into which process metaphors are active. Without perceptible cues it is impossible to tell exactly what the user can do next. Element metaphors can be graphics, sounds, text, touch and anything else that the user can perceived through the senses.

An example of this is the way in which the interface to a budgeting program looks like a ledger, with appropriate headings and so forth. This metaphor, THE INTERFACE IS A BUDGETING LEDGER cues the user into the fact that they can use the budgeting process metaphor to interact with the system. Similarly the paintbrush cursor used in a painting program cues the user into the process metaphor THE CURSOR IS A PAINTBRUSH and thus should help them to interact in the appropriate fashion. (Note that the use of an icon to enter the mode is an instance of what we might call an "element metonymy".)

Element metaphors generally correspond closely to things in the real world that we associate with the particular actions accepted by a process metaphor. To discover the appropriate representations we ought to think about the notion of *affordances* [6, 13] to help us to identify and assess the aspects of the real-world object that indicate what their process are. The theory of affordances is also directly related to how we

should reflect metaphoric entailments in an interface.

The process and element distinction in interface metaphors enables us to distinguish the perceivable and non-perceivable or functional aspects of a system and to approach them differently. Note also that process metaphors can be layered *over* existing functionality, rather than being designed in from the beginning, although this is not really desirable.

METAPHORIC ENTAILMENTS

The most important concept for formally understanding how metaphors function and will affect an interface design is the idea of *metaphoric entailments*. A metaphoric entailment is *a description of one thing the signifier implies about the signified* and is fundamental in Lakoff and Johnson's work.

Technically, any aspect of the signifier can give a metaphoric entailment. In conventional metaphors, the set of metaphoric entailments is fairly delineated and so people know which apply within the metaphor and which do not. When using a novel metaphor, however, the entailments are not firmly established and should be carefully indicated.

If we informally examine a common interface metaphor we can see some of the metaphoric entailments immediately. Suppose we consider USING THE DATA-STORAGE SYSTEM IS FILING. This has some entailments which are applicable:

- There are files in the data-storage system.
- There are folders in the data-storage system.
- Files can be placed in folders in the data-storage system.

But also some entailments which are not applicable:

- Folders can be placed in drawers.
- There is a filing cabinet in the data-storage system.

Before we move on, note that there is a subtle problem with the way this metaphor is actualised in user-interfaces. In the real world a file is usually a *folder* full of material. In the user-interface we *distinguish* between files and folders, thus breaking the metaphor to some degree. Most of us are so used to the *computerised* version of filing that we do not even notice, although some first-time users may well be confused by this very issue.

In order to use metaphoric entailments to successfully characterise and examine interface metaphors, we need some way of formally representing them. The most obvious way is to provide them in the form of deductive arguments, suggested by the fact they are known as *entailments*. An example from Lakoff and Johnson [8, p.91] :

> AN ARGUMENT IS A JOURNEY
> A JOURNEY DEFINES A PATH
> ————————————————————
> Therefore, AN ARGUMENT DEFINES A PATH

In this case the bottom line represents a metaphoric entailment, while the top two lines represent the deduction that reveals it. This is a useful way to think about metaphoric entailments because it presents us with the reasoning that establishes the entailment. It also allows us to see, in the sec-

ond premise, the aspect of the signifier that is used to gain the metaphoric entailment, in this case that "a journey defines a path". This makes it clear that we can potentially take *any* statement about the signifier and turn it into a metaphoric entailment, although this is not necessarily desirable.

Essentially we can see that metaphoric entailments are a matter of the transfer of knowledge about the signifier onto the signified to gain some understanding of the signified. In order to fully understand any user-interface metaphor, then, we must come to grips with a) the metaphoric entailments of the metaphor in general and b) which metaphoric entailments are applicable in the interface and which are not. This is not a simple problem, and will be briefly addressed in the heuristics section and in future work.

Before moving on, it is worth noting that the approach of semiotics [1] and also algebraic semiotics [7] are highly applicable in the case of metaphoric entailments. Indeed, they may well provide an excellent formal basis for discussing the structure of metaphor. More work needs to be done to create the exact linkages, but much of what is needed is already in place in the references cited.

SOME SUGGESTED HEURISTICS

As we examined the taxonomy of interface metaphor and the notion of metaphoric entailment in the previous sections we noted that some problems can arise when we use metaphor in interfaces. For some of these issues knowledge of metaphor leads to some suggestions for heuristics. These heuristics are strongly grounded in the taxonomy discussed and can be verified further through practical application.

The use of systematic heuristics has been comprehensively shown to be a useful technique for user-interface evaluation by Jakob Nielsen [11, 12]. In the heuristics below, the issues are not of general usability, but concern the understanding and use of interface metaphors. Following Nielsen, these heuristics are fairly informal and represent a common-sense approaches to issues.

Novel and Conventional Metaphors

1. *With conventional metaphors, ensure that you know the structure, and are able to clearly indicate which standard metaphoric entailments are not applicable in the interface.*

2. *With novel metaphors, ensure that all metaphoric entailments are clearly indicated.*

We obtain heuristic 1 very simply because, when using conventional metaphors, the metaphor structure already has an accepted form as noted in the taxonomy. If we deviate from this form in any way, it must be clear to the user we have done so. Otherwise, there will be a discrepancy between the user's understanding of the conventional metaphor and its actual implementation in the interface.

Heuristic 2 is a result of the nature of novel metaphor: the metaphoric entailments are not established in common understanding. As we cannot enter into a dialogue with the user to agree on the active metaphoric entailments we must indicate which are active through good cuing.

Entailment Selection

As seen in the taxonomy, metaphors can be characterised as collections of metaphoric entailments. Given that non-applicable metaphoric entailments will be confusing and frustrating to the user when interacting with a system:

3. *Use as many of the metaphoric entailments implied by metaphors as possible.*

The reason for this heuristic is clear: the more often the user can follow their knowledge of metaphoric entailments with success, the more they will trust the metaphor and the easier and more comfortable and consistent the system will seem. Part of this buy-in comes from the application of heuristics 1 and 2 also. It is important, though, not to use entailments simply for the sake of using as many as possible. All choices should be justified as explaining an aspect of functionality.

An example of the use of this heuristic can be seen in a book metaphor for documentation. Using the entailment "you turn the pages manually to find information" is a poor idea for a user-interface. To indicate it is not available, perhaps we could make some *other* means of navigation obvious. On the other hand, an entailment it would be *good* to use is "you can use the index to find specific information". This utilises the benefits of a computer to *improve* the experience of a book, while essentially keeping the process the same.

One option for actually identifying and displaying entailments is to use the affordance theory originally posited by J. J. Gibson [6] as discussed earlier. Donald Norman's application of this to object design is also instructive [13]. The notion of perceiving possibilities for action is just what we are looking for when speaking of how to establish which metaphorical entailments are active in a system. The existing theory on this as cited here provides the beginnings for a systematic approach to this problem.

Orientational Metaphors

As discussed in the taxonomy, orientational metaphors structure *across* multiple concepts and thus define entire frameworks of related concepts by giving them a spatial orientation. This leads us to the heuristic:

4. *When using orientational metaphors, make sure they fit into the conceptual framework defined by that orientation.*

The most important point here is that, when using orientational metaphors, we must be aware of the other members of the group they structure. For example, because GOOD IS UP we do not then want to associate an upward orientation with errors, but do want to associate that orientation with success messages. The key is to be aware of the implications inherent in every orientational metaphor and to judge whether the implications are appropriate for the interface metaphor.

Process Metaphors

The nature of process metaphors is to explain the functionality of the system in terms of real-world processes. As each process metaphor is quite complex in its own right, we obtain the heuristic:

5. *Use as few process metaphors as possible to gain good coverage of the system functionality.*

There is a natural trade-off here between metaphor coverage of functionality and the complexity of the metaphoric world defined by the interface. It is desirable to explain using metaphor, but it is also undesirable to force the user to recall too many different process metaphors. Additionally, using many process metaphors leads to what we might call 'entailment overload', where there are simply too many actions implied by the system, and the user is overwhelmed. The trade-off favours minimising process metaphors over maximising coverage.

Element Metaphors
In the taxonomy we have seen that element metaphors are very closely connected to the process metaphors of a system. In fact, their fundamental purpose is to indicate the presence of those metaphors. Therefore:

6. *Base every element metaphor on a process metaphor.*

We should never produce element metaphors that are unconnected with a process metaphor. In addition, each process metaphor should ideally have multiple element metaphors, as this gives the user multiple opportunities to recognise its presence in the interface. To do this, utilise all media elements available to create element metaphors that strongly point to the available process metaphor(s). Finally, heuristic 6 also emphasises that we should design element metaphors with the process metaphor very closely in mind so that we can make it as apparent as possible.

Culture/Groups
One thing we have seen in the taxonomy of metaphor is that metaphors are very sensitive to the metaphor-user's culture or group background. Because of this we get the heuristic:

8. *Seek to understand the users' metaphoric world as deeply as possible to counter the problem of the deeply subjective nature of metaphor.*

Specifically, we need to find which metaphors and metaphoric entailments the target audience has in common, as this will enable us to satisfy as many users as possible. This is connected with Jakob Nielsen's usability engineering principle of "know the user" [12], but tailored towards a specific property of the user group.

In fact, almost all metaphor are influenced by culture because of the way in which they are chosen. Which metaphors are selected depends largely on their coherence with the other metaphors in the culture's system, and therefore on the culture itself. This especially applies to orientational and ontological metaphors.

CONCLUSIONS
In this paper we have provided a principled understanding of user-interface metaphors along with a vocabulary for discussing and analysing them. We have also shown that the reasoning presented in the taxonomy could be the basis for some usability heuristics.

It should be clear from the discussion that some of the concerns raised by researchers on metaphor can be allayed by this more rigorous approach. In particular, concerns that metaphor is used too casually are immediately dealt with. Additionally, the charge that metaphor is an awkward tool because of misplaced entailments and so forth can now be properly assessed, and we are optimistic that solutions can be found. The more major criticism that metaphor is not a useful tool [2] is not directly dealt with here, but the analysis present in the taxonomy should indicate that there are many benefits to user-interface metaphors *if we choose them correctly and harness them properly*.

REFERENCES
1. Andersen, P. B. *A Theory of Computer Semiotics*, vol. 3 of *Cambridge Series on Human-Computer Interaction*. Cambridge University Press, 1997.

2. Blackwell, A. *Metaphor in Diagrams*. PhD thesis, University of Cambridge, September 1998.

3. Carroll, J. M., Mack, R. L., and Kellogg, W. A. Interface metaphors and user interface design. In *Handbook of Human-Computer Interaction*, M. Helander, Ed. Elsevier Science Publishers, 1988.

4. de Saussure, F. *Cours de linguistique générale*. Charles Bally et Albert Sechehaye, 1916.

5. Erickson, T. D. Working with interface metaphors. In *The Art of Human-Computer Interface Design*, B. Laurel, Ed. Addison Wesley, 1990.

6. Gibson, J. J. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1986.

7. Gougen, J. An introduction to algebraic semiotics, with application to user interface design. In *Computation for Metaphor, Analogy and Agents*, C. Nehaniv, Ed., vol. 1562 of *LNAI*. 1999, pp. 242–291.

8. Lakoff, G., and Johnson, M. *Metaphors We Live By*. The University of Chicago Press, 1980.

9. Laurel, B. *Computers as Theatre*. Addison-Wesley Publishing Company, Inc., 1993.

10. Mandel, T. *The Elements of User Interface Design*. John Wiley and Sons, Inc., 1997.

11. Nielsen, J. Heuristic evaluation. In *Usability Inspection Methods*, J. Nielsen and R. L. Mack, Eds. John Wiley and Sons Inc., 1993.

12. Nielsen, J. *Usability Engineering*. Academic Press, 1993.

13. Norman, D. A. *The Design of Everyday Things*. Doubleday, 1988.

14. Wozny, L. A. The application of metaphor, analogy, and conceptual models in computer systems. *Interacting with Computers 1*, 3 (1989).