

# A Taxonomic Analysis of User-Interface Metaphors in the Microsoft Office Project Gallery

Pippin Barr<sup>1</sup>

Rilla Khaled<sup>1</sup>

James Noble<sup>1</sup>

Robert Biddle<sup>2</sup>

<sup>1</sup>Victoria University of Wellington, New Zealand

<sup>2</sup>Carleton University, Canada

Email: [chikken@mcs.vuw.ac.nz](mailto:chikken@mcs.vuw.ac.nz)

## Abstract

User-interface metaphors are not well understood in terms of their relationships and qualities. They are, however, constantly used both consciously and unconsciously in most user-interface designs. This paper demonstrates a taxonomic classification and analysis of the user-interface metaphors in the Microsoft Office Project Gallery. The classification offers insight both into the nature of the metaphors in the user-interface investigated and more generally, and helps validate the use of the taxonomy as an assessment tool.

*Keywords:* user-interface, metaphor, taxonomy

## 1 Introduction

Metaphors are a very popular approach to user-interface design. Two of the largest interface design companies in the world, Apple and Microsoft, both strongly recommend the use of metaphor:

You can take advantage of people's knowledge of the world around them by using metaphors to convey concepts and features of your application. Use metaphors involving concrete, familiar ideas and make the metaphors plain, so that users have a set of expectations to apply to computer environments.

(Apple Computer, Inc. Staff 1992)

Familiar metaphors provide a direct and intuitive interface to user tasks. By allowing users to transfer their knowledge and experience, metaphors make it easier to predict and learn the behaviors of software-based representations.

(Microsoft Corporation 1995)

There has been considerable research into user-interface metaphor. Some work aims to emphasise the potential benefits of the technique (Carroll, Mack & Kellogg 1988, Johnson 1997). Other research and commentary has focused on key problems with its use (Kay 1990, Nelson 1990, Norman 1998). Finally,

Copyright ©2005, Australian Computer Society, Inc. This paper appeared at the 6th Australasian User Interface Conference (AUI2005), Newcastle. Conferences in Research and Practice in Information Technology, Vol. 40. M. Billingham and A. Cockburn, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

This work was supported by a New Zealand Top Achiever Doctoral Scholarship.

there has been considerable practical advice given on the use of metaphor (Halasz & Moran 1982, Erickson 1990, Madsen 1994). Despite this fairly large output, there has been little investigation into the underlying concept itself, and this has left discussion somewhat unguided and relying heavily on intuition.

In their book *Metaphors We Live By*, George Lakoff and Mark Johnson (1980) present a systematic examination of the use of metaphor in everyday life, including a number of categories of metaphor. In a previous paper we presented a taxonomy of user-interface metaphor based on and extending this work (Barr 2002). Barr (2003) then examined the taxonomy in depth and performed a large amount of practical work concerning the analysis of user-interface metaphors individually and in collection. This paper will present the application of the taxonomy to the Microsoft Office "Project Gallery" (Corporation 2001). In doing this, we will show that thinking in a structured and categoric way about the metaphors used in the interface yields considerable insight and value.

In section 2 we briefly present our adaptation of the Lakoff and Johnson taxonomy to user-interface metaphor. Following this we explain how the study of the Project Gallery was performed and describe four specific examples from it as a means to demonstrate specific insight gained from use of the taxonomy (section 3). In section 4, we go on to describe some more general factors revealed by the analysis of the interface, and suggest that these might very well generalise. Finally, we discuss related and possible future work in section 5, before concluding in section 6.

## 2 A Taxonomy of User-Interface Metaphors

The taxonomy of user-interface metaphor shown in figure 1 is presented here in fairly abbreviated form for reference. It is additionally restated to reflect certain emphases of this paper. This includes the backgrounding of metaphorical entailments (discussed below), which have become more central to our work in semiotics, and are discussed in far more detail there.

For the purposes of this paper, a user-interface metaphor is defined as a device for explaining some system functionality or structure (the *tenor*) by asserting its similarity to another concept or thing already familiar to the user (the *vehicle*). The traditional form of expressing a metaphor is the statement  $X$  is  $Y$ ,<sup>1</sup> where "X" is the tenor, and "Y" is the vehicle.

After briefly introducing the concept of metaphorical entailments, which play a key role in the analysis of user-interface metaphors, we will proceed to describe the taxonomic categories to be considered in

<sup>1</sup>We follow Lakoff and Johnson (1980) in the use of a small-caps font for the statement of metaphors.

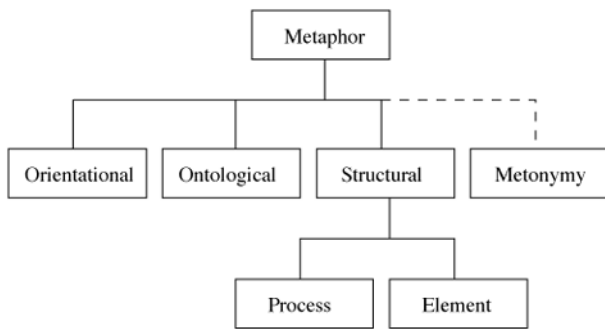


Figure 1: A Taxonomy of User-Interface Metaphor, from (Barr 2002).

the case-study below. For each category or concept we seek to give a brief description of its nature, followed by a rationale for its use, any common misuses of the concept, and finally one or two examples to clarify.

### 2.1 Metaphorical Entailments

The metaphorical entailments of a user-interface metaphor are a mapping of facts about the *vehicle* (the real world concept being used to do the explaining) onto the *tenor* (the system concept being explained). Thus, the metaphor THE INTERFACE IS A WIZARD, used in the common “wizard” interface of many applications, has metaphorical entailments such as “*the interface knows more than I do*” and “*the interface uses a wand.*” The complete set of metaphorical entailments is not necessarily implemented in an actual user-interface, but rather a (hopefully) suitable subset deemed useful by the designer(s).

As an extended example, consider the metaphor THE DATA IS A DOCUMENT, which has a multitude of possible entailments. Given that there are so many, a full list will not be presented here. Instead, we present a subset of both potentially useful and potentially useless entailments to give an idea of the range:

- The data is made of paper with ink on it.
- The data can be crumpled into a ball and thrown into the trash.
- The data consists of words, figures, and images.
- The data contains information.
- The data can be torn, or have coffee spilled on it.
- The data has a title.
- The data can have footnotes, a bibliography, and so on.

As can be seen, any fact about real-world documents can be transferred to apply to the concept in need of explanation. Examining metaphorical entailments in a systematic way is a valuable means to assess and design user-interface metaphors.

### 2.2 Orientational Metaphor

An orientational metaphor is one that maps an interface concept onto a spatial concept such as “up”, “down,” “left,” or “right.” Orientational metaphors



Figure 2: An orientational metaphor.

leverage everyday understanding of spatiality in order to convey useful information, especially concerning navigation, quantification, and priority.

As an example of an orientational metaphor, consider the common “up” and “down” controls used to indicate a quantity (figure 2). In this case, clicking the “up” arrow will cause the quantity to increase. Thus there is a metaphoric association of “up” with “more” and “down” with “less.” Lakoff and Johnson (1980) suggest this kind of association likely relates to the idea that a pile of substance gets higher the more you add, growing upward.

One issue surrounding this form of user-interface metaphor is the fact that different cultures can have substantially different associations with spatial concepts. Additionally, little thought is given to the specifically metaphoric nature of orientation in the interface.

### 2.3 Ontological Metaphor

An ontological user-interface metaphor is one that identifies a system concept with a basic category of existence in the physical world, such as substance, object, container or entity. Ontological metaphors allow users to make use of their basic knowledge of how the world physically functions to negotiate the user-interface, especially for the purposes of referral, quantification or indication of causation. Because they are so basic, ontological metaphors are often used unknowingly, and can generate entailments that are undesirable within the function of the system.



Figure 3: An ontological entity metaphor.

As an example of an ontological metaphor being used to indicate causation, consider figure 3 above. In this case we have the ontological metaphor THE FRONTPAGE APPLICATION IS AN ENTITY which allows the interface to describe a causal event. By presenting the application as an entity capable of “needing” something, the desired result can be achieved. Other

common ontological metaphors include THE COLLECTION OF DATA IS AN OBJECT (a file) and THE FILE IS A CONTAINER (a folder).

## 2.4 Structural Metaphor

A structural metaphor is one that identifies an abstract system concept with a detailed real world concept or object. This identification is intended to be useful in that the real world concept or object has a large degree of structure and thus potentially provides considerable prior knowledge for the user to leverage.

As an example of a structural metaphor, take for example the trashcan from MacOS X shown in Figure 4. It illustrates the FILE DELETION IS USING A TRASHCAN metaphor, which associates the system concept of file deletion with the real world concept of throwing unneeded items into the trash. Other examples of structural metaphors include music players, budgeting programs, toolbars, documents, and wizards.

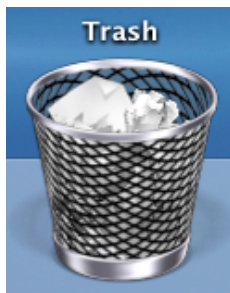


Figure 4: A structural metaphor.

An important issue with structural metaphors is that the transfer of facts about the real world object to system concept (in the form of metaphorical entailments) is not a complete one and may lead to user confusion. This problem can be compounded further if an unsuitable real world concept is chosen as the representation.

## 2.5 Conventional and New Metaphor

A conventional metaphor is one that is already familiar to users, who can then expect certain behaviour upon encountering it. A new user-interface metaphor is one that is not conventional, and hence has not (yet) become familiar to users. Consequently, they are not well understood and users will often not be sure which metaphorical entailments are applicable.

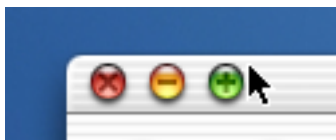


Figure 5: A new metaphor.

As an example of a new metaphor, consider the window control buttons in MacOS X, shown in figure 5. These buttons use the metaphor THE WINDOW CONTROLS ARE A TRAFFIC LIGHT to explain window functions in terms of traffic lights and how they control cars. The metaphorical entailments such as “red

means stop” (close the window) are not necessarily well known or accepted by its users. Apple therefore tries to clarify the entailments to the users by including symbols which appear when the user moves their mouse over the window controls, as shown in the figure.

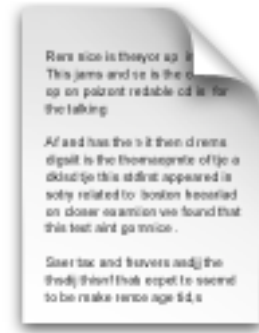


Figure 6: A conventional metaphor.

In contrast, one of the most conventional metaphors in modern day user-interfaces is the document metaphor seen in figure 6 (THE COLLECTION OF DATA IS A DOCUMENT). The various metaphorical entailments of this user-interface metaphor are quite well established and users generally know exactly what to expect, and, equally importantly, what not to expect when interacting with the metaphor. Hence, users expect a document to contain data, and especially text, but not necessarily to be written out on paper.

An issue surrounding the use of conventional metaphors is that, as noted, users have certain expectations of how they will work, and unless the designer uses all of the expected metaphorical entailments, user confusion could result.

## 3 The Microsoft Office Project Gallery

We used the taxonomy to perform an analysis of the Microsoft Office “Project Gallery” included as part of Microsoft Office for MacOS X (figure 7). The Project Gallery is designed to appear as soon as a user launches any of the Microsoft Office applications, such as Microsoft Word or PowerPoint. Additionally, the Project Gallery is the first option presented under the “File” menu of these applications. These facts indicate that the Project Gallery user-interface is intended to be presented to all users the first time they launch an Office application, and that it is presumably aimed at beginner users. Its basic purpose is to “get users started” with their tasks by providing a collection of templates and wizards for common needs.

To apply the taxonomy we first identified seventeen different metaphors immediately apparent in this one small user-interface. The metaphors found are listed in figure 8. These metaphors were located without much difficulty and this serves to confirm the claim made in the introduction that user-interface metaphors are very common in today’s user-interfaces. Having identified the metaphors we proceeded to categorise and analyse them according to the taxonomy. This involved several steps: expressing the metaphor involved in the traditional form of “X IS Y”; discovering all of the ways it was represented in the interface; analysing its use in the user-interface; classifying it according to the three cate-

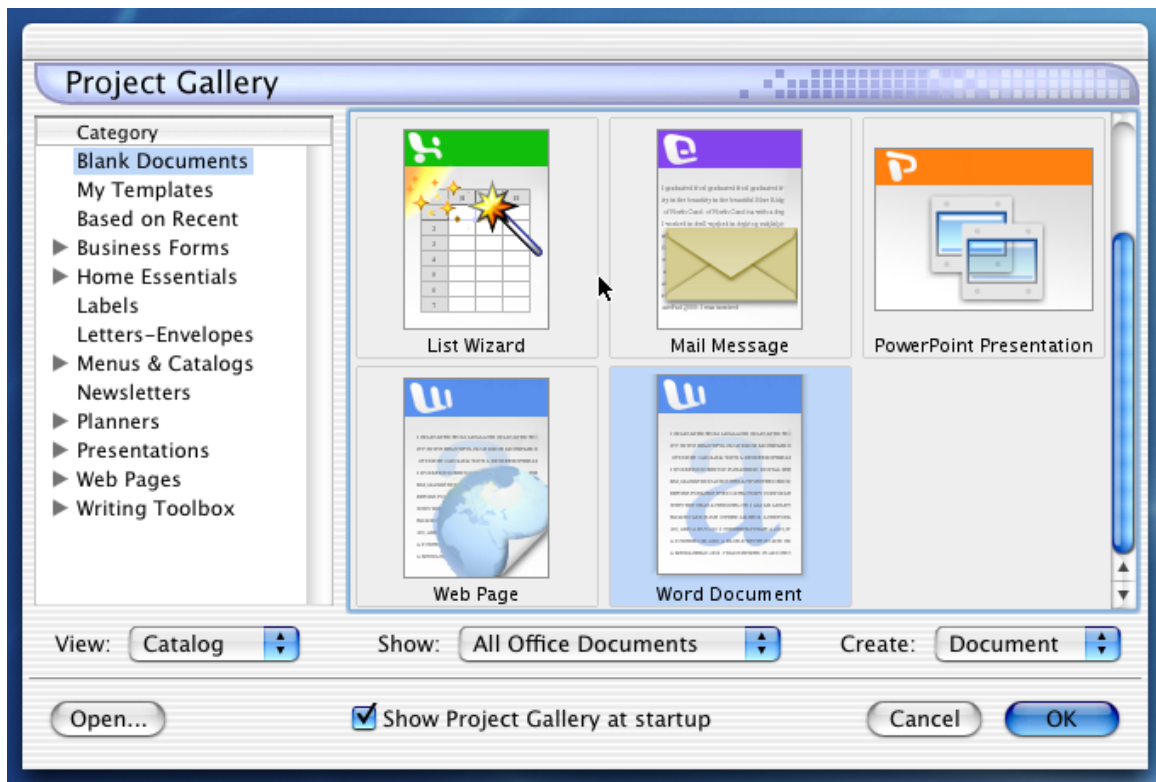


Figure 7: The Microsoft Office “Project Gallery”.

gories of metaphor as discussed above and also as novel or conventional; and, finally, noting any insight gained via the classification.

There is not room enough in this paper to analyse all seventeen metaphors, so we instead provide four major examples of metaphor use in the Microsoft Office Project Gallery. This will be followed by some general patterns observed during the analysis. We note here, before presenting the examples, that they are all structural, and largely conventional in nature. The prevalence of structural and conventional metaphors is one of the many general findings we obtained from performing the case-study.

### 3.1 The Gallery Metaphor

**Metaphor** THE COLLECTION OF TEMPLATES AND WIZARDS IS A GALLERY

**Representation** The gallery metaphor is indicated via language in the title-bar of the window which states that it is the “Project Gallery” (figure 7). It is additionally represented through graphical means, as the interface itself resembles the wall of a gallery in that there are pictures with captions or labels set against a white background (figure 7). Finally, the gallery metaphor is reinforced by the presence of the catalog metaphor, which is also present in the interface, as seen in the text beside the “View” option (again in figure 7).

**Usage** In real life a gallery is a building which contains largely visual media. Art in a gallery is often organised into one or more “shows,” which contain art with some common feature (such as the artist or subject). Patrons of a gallery are invited to look at, but generally not to touch, the art on display. In the user-interface the gallery is intended as a container of certain templates and

wizards. Each of these is visually represented in the format of pictures hanging on a wall, as mentioned above. They are organised into categories, which seem to correspond to the concept of shows in a gallery. The metaphor is therefore an overarching metaphor for the interface to explain why all of these items are in the same place.

**Classification** The gallery metaphor is a structural one, as galleries tend to have fairly well-defined basic features in the real world. However, it is worth noting that, other than the use of a visual presentation similar to that of a gallery, the metaphor seems to function more as an ontological and generic container metaphor, a place to store things. The gallery metaphor seems to be relatively new one and would potentially surprise a user if it were put forward with any more force.

**Comments** Although the gallery metaphor is a structural one, it is easy to see it is very close to being an ontological container metaphor. This is because the implementation in the interface involves very few of even the most obvious and common entailments of a gallery. For example there is no obvious inclusion of the concepts of “art” or “exhibitions.” What is more, the absence of so many of the fundamental entailments of a gallery could cause potential problems for a user who either expects or assumes them to be there. One problem could be an assumption that the contents of the gallery are valuable and not to be touched, but only looked at. Additionally, the gallery metaphor entails several roles, including “director,” “curator,” and “patron.” It is not made clear which role the user is intended to play, though we might assume that they are the director of their own gallery in this case. Finally, we can note from the analysis of

<i>Metaphor</i>	<i>Classification</i>
THE COLLECTION OF TEMPLATES AND WIZARDS IS A GALLERY	Structural and Conventional
THE VIEW OF THE COLLECTION OF PROJECTS IS A CATALOG	Structural and Conventional
THE COLLECTION OF TEMPLATES IS A TOOLBOX	Structural and Conventional
THE TEMPLATE IS A TOOL	Structural/Ontological and Conventional
THE PRE-FORMATTED DOCUMENT IS A TEMPLATE	Structural and Conventional
THE INTERACTION PROCESS IS A DIALOG	Structural and Conventional
THE DELIMITED AREA ON THE SCREEN IS A BOX	Structural/Ontological and Conventional
THE COLLECTION OF DATA IS A DOCUMENT	Structural and Conventional
THE DIALOG BOX(ES) IS A WIZARD	Structural and Conventional
THE COLLECTION OF SOFTWARE IS AN OFFICE	Structural and Conventional
THE INTERNET IS A WEB	Structural and Conventional
THE COLLECTION OF DATA IS A PAGE	Structural and Conventional
THE RECTANGULAR AREA ON THE SCREEN IS A WINDOW	Structural and Conventional
THE SUBSECTION OF THE WINDOW IS A PANE	Structural and Conventional
THE INFORMATION TRANSFER IS MAIL	Structural and Conventional
THE SOFTWARE IS AN ENTOURAGE	Structural and New
THE AREA OF THE WINDOW IS A SCROLL	Structural and Conventional

Figure 8: Metaphors found in the Microsoft Office “Project Gallery”.

the gallery metaphor that it actually serves to entail another metaphor in the system, namely the catalog metaphor (not examined here). This concept is an interesting one, and will be discussed in section 4.

### 3.2 The Toolbox Metaphor

**Metaphor** THE COLLECTION OF TEMPLATES IS A TOOLBOX

**Representation** The toolbox metaphor is represented solely via language in the category heading “Writing Toolbox” as shown in figure 7.

**Usage** In the real world, a toolbox is used as a place to store tools and often contains some means of categorising them within it. It is intended as a device to carry a particular set of tools around so that they are always available to their user. It is possible to have specialised toolboxes, depending on the sorts of tasks that need to be performed, such as a “carpenter’s toolbox” or a “plumber’s toolbox.” In the Project Gallery, the toolbox metaphor is used to explain the collection together of a particular set of templates. In this case it is a collection of templates specifically used for “writing,” and includes different kinds of templates for writing either “Journals” or “Reports.” Within these categories are templates for documents such as a “Baby Journal,” and a “Book Report.”

**Classification** The toolbox metaphor is essentially a structural one. Real world toolboxes involve a particular base set of characteristics which are being transferred to the category in the Project Gallery. The toolbox concept is a fairly conventional one in user-interfaces, along with the similar metaphor of a “toolkit.”

**Comments** Because the toolbox metaphor is conveyed only via language it does not involve any of the visual entailments we might expect, such as its general appearance of being a metal box with a handle and so on. This serves to focus the entailments solely on the toolbox’s functional role of storing and categorising tools. Because of this, the metaphor is, like the gallery

metaphor, quite close to being an *ontological container metaphor* instead, and is really a minor specialisation of this. We note that, because the toolbox metaphor is a fairly common one, its lack of visual representation is unlikely to cause confusion in any but the most novice of users.

### 3.3 The Tool Metaphor

**Metaphor** THE TEMPLATE IS A TOOL

**Representation** The tool metaphor is represented via language, in the name of the toolbox. It is additionally entailed by the toolbox metaphor, which can be considered a form of representation.

**Usage** In real life a tool is generally some object which can be used to achieve one or more specific tasks, such as a screwdriver. Additionally, we sometimes refer to non-physical things as tools, such as “programming tools” (compilers, programming environments, etc.) and “theoretical tools” (algebra, calculus, and so on).

In the user-interface the tool metaphor is used to help explain the purpose of the templates stored inside the “Writing Toolbox.” By thinking of these digital templates as “tools,” the user is informed that they are specialised elements of the program to be used for solving particular tasks.

**Classification** The tool metaphor can be thought of as structural, because it refers to a kind of real-world object. However, it is also borderline ontological, because the concept of a tool is quite basic within human cognition. The tool metaphor as regards user-interface is quite a conventional one, and most users will likely feel little surprise at its usage here.

**Comments** The tool metaphor is entailed by the toolbox metaphor, rather than being explicitly represented in the user-interface. This lack of direct representation serves to decrease the importance of the metaphor in the system to the point where it is unlikely a user will be aware of its presence. This appears to be quite a common property of ontological metaphors, which generally escape explicit notice.

### 3.4 The Template Metaphor

**Metaphor** THE PRE-FORMATTED DOCUMENT IS A TEMPLATE

**Representation** The template metaphor is represented via language and graphics. The term “template” is used throughout the Project Gallery to refer to documents. Examples include the “My Templates” category (figure 7), and the phrase “The Project Gallery contains templates you can use” in a description of the Project Gallery. Additionally, the icons representing the various templates in categories such as “Business Forms” present a preview of the template by showing how it looks before it is filled out. This presentation of a document with some layout already present is similar to the appearance of a physical template.

**Usage** In real life a template is generally a thin sheet of metal or some other substance which is used to guide drawing or lettering of some kind. Templates are intended to make repetitive work both easier and more consistent by providing a standard interface to the task.

In the user-interface the template metaphor is used to indicate that a certain kind of document has been pre-formatted and has already existing content to help the user achieve a task with consistent results across attempts. The template serves as a “head start” on a particular task by, in some sense, doing some of the work for the user.

**Classification** The template metaphor is structural, because real templates do exist and serve a particular purpose. Additionally, the template metaphor is a conventional one as the template concept has been around in user-interfaces for some time now. It is fairly common for the word “template” to call to mind a pre-formatted computer document before anything else, in fact.

**Comments** The template metaphor is interesting in that it shows how fairly abstract concepts can be used as structural metaphors. Although templates can be real-world objects, in conversation there would be a high chance that the first kind of “template” a person thought of would in fact be a computer-based one. This suggests that the template metaphor has achieved a significant penetration into the way that people think about the term itself. Also noteworthy is the pleasant correspondence between the suitability of finding real world templates in a toolbox, and digital templates in a digital toolbox. These sorts of relationships would seem to strengthen the overall metaphoric nature of the interface.

## 4 Discussion

In the course of analysing the seventeen user-interface metaphors identified, we found a number of interesting patterns. We present each of them here with some indication of their prevalence, along with an acknowledgement of exceptions and an attempted explanation for the phenomenon. Finally, we note that the Project Gallery is a fairly standard user-interface, and that many of these observations could well apply to user-interfaces more generally.

### 4.1 Dominance of structural metaphors

Of the seventeen metaphors identified, all seventeen could quite reasonably be considered structural. Two

potential exceptions were the tool metaphor and the template metaphor, both of which it was somewhat tempting to classify as ontological. Additionally, it is quite feasible to claim that there is an orientational metaphor at work in the scroll metaphor involved in the scrollbar.

The reason for this predominance of the structural metaphor is fairly clear: a structural metaphor provides a maximal amount of detail which can be applied to the user-interface concept. That is, there is simply more information to leverage in order to help the user’s understanding. Additionally, it is far easier to represent structural metaphors in a user-interface, again because of their higher degree of specificity. Finally, we note that most advice on the use of metaphors tends to emphasise structural metaphors as being the most desirable due to their likelihood of being more obviously familiar (see, for example the Apple (1992) guideline quoted in the introduction).

### 4.2 Dominance of conventional metaphors

Of the seventeen metaphors identified, only one or two might be considered truly novel. Both the gallery metaphor, discussed above, and the entourage metaphor, which applies to the calendar program of the same name, seem relatively uncommon. The entourage metaphor is so unusual as to quite probably have been used only with reference to this particular application, and seems to leverage the metaphorical entailments of having an entourage to take care of one’s daily schedule, email, address book, and so on.

This result is not terribly surprising. Microsoft Office has been around for some time now, and the set of metaphors it uses have thus become quite familiar to us. Additionally, the Project Gallery is intended for new users, and it is possible that a design decision to limit the number of new metaphors might have been taken. Finally, Microsoft is a large and fairly conservative company, and thus unlikely to use too many metaphors whose novelty could potentially confuse their users.

### 4.3 The difficulty of differentiating structural and ontological metaphors

A point raised by the attempt to classify metaphors such as tool and template is that it is not always straightforward to state precisely whether a metaphor is structural or ontological. This suggests that there may be something of a continuum between the two categories, rather than a clear division as we had originally thought. Ontological metaphors are fairly abstract characterisations of the basic elements making up the world, while structural metaphor are more specific. This level of specificity is what would determine a metaphor’s place on the continuum. Thus, a container metaphor might be further specified as a box metaphor, then as a toolbox metaphor, and finally as a carpenter’s toolbox metaphor, gaining detail as it moves along this path.

### 4.4 The difficulty of differentiating new and conventional metaphors

Following on from the above, we observed that it was difficult at times to precisely classify a metaphor as being specifically new or conventional. Instead, the instances seemed to fall along a continuum. Thus, for example, the wizard metaphor is more conventional than the gallery metaphor, but probably less so than the document or template metaphors.

A second key insight gained from this classification was that it seemed some user-interface metaphors

were so conventional as to be simply alternative definitions of the term itself. Thus, for example, it is often the case that if one talks about “templates” in day-to-day life, it is quite ordinary for people to assume the reference is to a template on a computer. We call such ingrained user-interface metaphors definitive. These two observations suggest the existence of a “life-cycle” of user-interface metaphor. Any metaphor newly introduced begins, more or less, at the extreme of novelty. As it is used, it gradually becomes more and more conventional. Conditional on its continuing use, the metaphor may eventually become definitive. Reaching the definitive state is presumably desirable because it is at this point that the metaphor blends into the interface seamlessly. We must note, however, that this is not necessarily because the metaphor is good, but possibly because it is simply used all the time and users have learned its behaviour.

#### 4.5 The use of ontological metaphors

It was observed that most of the metaphors were either borderline ontological or based on ontological metaphors. Thus, for example, the ontological container metaphor forms the basis for such structural metaphors as gallery, toolbox, and dialog box. The ontological object metaphor clearly forms the basis of most of the structural metaphors, as they are based on physical concepts, and hence generally reflect some object-like qualities. This observation conforms with the nature of modern-day graphical user-interfaces and the popularity of “direct manipulation” (Shneiderman 1997). In fact, it is arguable that ontological metaphors provide the key entailments of many of the metaphors in the Project Gallery, such as that one can “put things into” the Gallery, or the Writing Toolbox (container metaphor), and that one can directly manipulate the Templates and Documents available (object metaphor).

#### 4.6 Entailment selection largely functional

In all metaphors identified, the metaphorical entailments were essentially functional. That is, they largely pertained to functionality available in the interface. There were admittedly one or two exceptions to this, such as the visual layout of the window as a catalog or gallery wall being entailed by those respective metaphors. On the whole, however, the entailments related to the metaphoric function of the interface element, rather than any other properties.

A reason for this could be that software is function-driven by its nature. Entailments which help to explain these functions are therefore likely to be the most useful. Additionally, if too many non-functional entailments were included, such as those concerning visual appearance or auditory consideration, the interface would be exceedingly confusing. Again, Microsoft is probably more likely than other software companies to be conservative in such choices.

#### 4.7 Entailment of other metaphors

An interesting observation made during our analysis was that the metaphorical entailments of a metaphor could entail another metaphor. Further, sometimes entailed metaphor was actually implemented in the user-interface. Thus, for example, the toolbox metaphor *implies the existence* of a tool metaphor, and the gallery metaphor *implies the existence* of a catalog metaphor. It must be noted, of course, that entailed metaphors were generally left out of the interface. There is no “security guard” metaphor as entailed by the gallery metaphor, for example. The potential for metaphors in an interface

entailing other metaphors is an interesting one, with each metaphor strengthening the representation and interpretation of the other, creating a strong coherence between them. Finally, we note that this use of entailed metaphor indicates a valuable means of discovering further metaphors for potential use in a user-interface.

#### 4.8 The potentially problematic omission of entailments

Every user-interface metaphor implemented in the Project Gallery omits various metaphorical entailments. For example, the office metaphor omits any physical setting, the catalog metaphor does not include the notion of turning pages, and the entourage metaphor does not include any representation or suggestion of a group of people. Clearly, there are some entailments which should be omitted, and the above are likely examples of these.

Omitting entailments of metaphors in a user-interface can, however, be problematic in at least two ways. First, it can confuse a user who expects an entailment to be implemented, and finds that it is not. A classic example of this comes in the wizard metaphor. Although a wizard in “real” life is a person, and thus subject, at least to some degree, to rules of social interaction, there is no social consideration given in the user-interface. In this case, the omission of the “social being” entailment seems to be a mistake, as research has shown that users expect to interact socially with computers, especially those espousing some sort of social agent role (Reeves & Nass 1996). The second problem of omitted entailments is that a user might think an entailment is present when it is not. Thus, for example, the gallery metaphor entails that the contents of the Project Gallery are works of art, and therefore should not be touched or mistreated. This could lead the user into a “look, but don’t touch” mode of interaction, which is the opposite of the intent of the interface.

#### 4.9 Physical metaphors rather than conceptual metaphors

Almost all of the metaphors discovered in the Project Gallery had a physical nature. The only potential exceptions were the dialog metaphor of the “dialog-box” and the mail metaphor implied by creation of “mail messages.” Each of these metaphors, and especially the former, can be said to deal with a conceptual process, rather than a physical object.

This discovery is interesting because we might expect more process-oriented metaphors than physical metaphors, given the functional nature of an interface. The answer, of course, is contained in the selection of functional entailments as already discussed above. Rather than having explicitly process-oriented metaphors, which can be hard to represent in an interface, physical metaphors with associated processes are used instead. We also note that user agreement on the properties and nature of physical objects is likely to be far higher than agreement as to the nature and meaning of abstract concepts such as “war” or “love.”

#### 4.10 Mixing metaphors

An interesting result of our case-study was observing the sheer proliferation of metaphors in the user-interface which could be said to be “mixed” or to not naturally work together. While, for example, we might be able to make some case for a “toolbox” being in a “gallery,” it is hardly plausible to encounter a “wizard” there. Similarly, the web metaphor does

not seem appropriate in the context of an office. The conventionality of many of these metaphors is what allows them to coexist without any surprise from the user, along with the generally unusual nature of user-interfaces. Users have simply got used to it. The mixing of metaphors cannot just be brushed off, however, and the co-existence of the office, gallery, and wizard metaphors, for example, could potentially trouble newer users of the interface. An instruction to “just get used to it” is not satisfactory. Considerable thought must be given to the coherence of an interface’s metaphors, and mixing metaphors can be destructive to it.

## 5 Related and Future Work

There are numerous approaches to the design of user-interface metaphors in the literature. Carroll et al. (1988) introduced one of the first systematic approaches to a design process, suggesting a sequence involving: metaphor generation, identification of potential mismatches between the metaphor and the system, and the management of those mismatches. In a similar vein, Erickson (1990) has provided an approach based largely on advice of how to generate potential explanatory metaphors. Madsen (1994) has developed an extensive set of heuristic guidelines concerning the generation, evaluation, and development of metaphors based the analysis of several case-studies. Finally, Smyth et al. (1995) have produced a set-based means to analyse particular metaphors’ explanatory power to improve their evaluation.

Our research group has undertaken extensive research in the area of metaphor analysis and classification, fitting in well with existing heuristic advice on metaphor generation. This research includes a detailed semiotic model of user-interface metaphor which utilises the concept of metaphorical entailments at a more formal level than has been the case in this paper (Barr, Noble & Biddle 2003). In another work we explore the concept of the ontological metaphor in considerable more depth and suggest that ontological metaphors may form the bedrock of modern graphical user-interface design (Barr, Biddle & Noble 2003b). We have additionally performed a heuristic evaluation of some of the user-interface metaphors used in music software (Duignan, Biddle, Noble & Barr 2004). Our most recent work has involved extending our semiotic approach to metaphor by applying it to the System Metaphor practise from Extreme Programming (Khaled, Barr, Noble & Biddle 2004a, Khaled, Barr, Noble & Biddle 2004b).

Finally, the concept of metonymy, although emphasised in our past works, has been left out of this paper as it has found a more precise embodiment in the semiotic concept of an “indexical sign.” For more on indexical/metonymic signs, see our work on the semiotics of user-interface sign categories (Barr, Biddle & Noble 2003a).

Future work we intend to perform includes presenting more detailed and well-established set of heuristics than were originally presented, and performing further heuristic evaluations. We additionally plan to show how we have used our semiotic model to analyse user-interface metaphors and to gain insight into their pros and cons in specific situations.

## 6 Conclusion

In this paper we have shown how the classification of metaphors in the Microsoft Office Project Gallery using the taxonomy of user-interface metaphors as described in this and other work has yielded insight both into the application and the taxonomy itself.

This insight has occurred both at the level of individual metaphors in the system, as well as in the form of general observation about the use of metaphor in the context of the taxonomy. We have also suggested that the results of our case-study could be generalised to similar sorts of user-interfaces. This paper has additionally shown that research into user-interface metaphor, despite being somewhat abandoned in recent times, still has considerable value, especially as metaphor is still heavily used, but taken more and more for granted in interface design.

## References

- Apple Computer, Inc. Staff (1992), *Macintosh Human Interface Guidelines*, Addison-Wesley.
- Barr, P. (2002), A taxonomy of user-interface metaphor, in S. Jones & M. Masoodian, eds, ‘Proceedings SIGCHI-NZ Symposium on Computer-Human Interaction’, The New Zealand Chapter of ACM SIGCHI.
- Barr, P. (2003), User-interface metaphor in theory and practice, MSc. Thesis, Victoria University of Wellington, Department of Mathematical and Computing Sciences.
- Barr, P., Biddle, R. & Noble, J. (2003a), Icons R Icons, in R. Biddle & B. Thomas, eds, ‘User Interfaces 2003: Fourth Australian User Interface Conference’, pp. 25–32.
- Barr, P., Biddle, R. & Noble, J. (2003b), Interface ontology: Creating a physical world for user-interfaces, in ‘Proceedings of Eighth European Conference on Pattern Languages of Programs’.
- Barr, P., Noble, J. & Biddle, R. (2003), A semiotic model of user-interface metaphor, in ‘Virtual Distributed and Flexible Organisations’, Kluwer Academic Publishers. Edited proceedings of the Sixth International Workshop on Organisations Semiotics in Reading, UK.
- Carroll, J. M., Mack, R. L. & Kellogg, W. A. (1988), Interface metaphors and user interface design, in M. Helander, ed., ‘Handbook of Human-Computer Interaction’, Elsevier Science Publishers, pp. 67–85.
- Corporation, M. (2001), ‘Microsoft Office X for Mac’.
- Duignan, M., Biddle, R., Noble, J. & Barr, P. (2004), Metaphors for electronic music production in live and reason, in ‘Proceedings of 6th Asian Pacific Conference on Computer Human Interaction (APCHI 2004)’.
- Erickson, T. D. (1990), Working with interface metaphors, in B. Laurel, ed., ‘The Art of Human-Computer Interface Design’, Addison-Wesley Publishing Company, pp. 65–73.
- Halasz, F. & Moran, T. P. (1982), Analogy considered harmful, in ‘Proceedings of the Conference on Human Factors in Computing Systems’, pp. 383–386.
- Johnson, S. (1997), *Interface Culture: How New Technology Transforms the Way We Create and Communicate*, Harper SanFrancisco.
- Kay, A. (1990), User interface: A personal view, in B. Laurel, ed., ‘The Art of Human-Computer Interface Design’, Addison-Wesley Publishing Company, pp. 191–207.

- Khaled, R., Barr, P., Noble, J. & Biddle, R. (2004a), Extreme programming system metaphor: A semiotic approach, *in* 'Proceedings of the 7th International Workshop on Organisational Semiotics'.
- Khaled, R., Barr, P., Noble, J. & Biddle, R. (2004b), Metaphorscape: Patterns for xp system metaphors, *in* 'Proceedings of the Ninth European Conference on Pattern Languages of Programs'.
- Lakoff, G. & Johnson, M. (1980), *Metaphors We Live By*, The University of Chicago Press.
- Madsen, K. H. (1994), 'A guide to metaphorical design', *Communications of the ACM* **37**(12), 57–62.
- Microsoft Corporation (1995), *The Windows Interface Guidelines for Software Design: An Application Design Guide*, Microsoft Press.
- Nelson, T. H. (1990), The right way to think about software design, *in* B. Laurel, ed., 'The Art of Human-Computer Interface Design', Addison-Wesley Publishing Company, pp. 235–243.
- Norman, D. A. (1998), *The Invisible Computer*, The MIT Press.
- Reeves, B. & Nass, C. (1996), *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*, Cambridge University Press.
- Shneiderman, B. (1997), *Designing the User Interface : Strategies for Effective Human-Computer Interaction*, 3rd edn, Addison-Wesley.