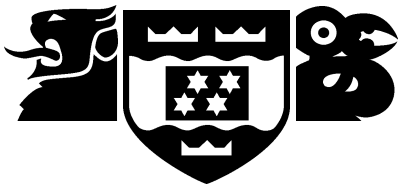
VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

Icons R Icons: User interface icons, metaphor and metonymy

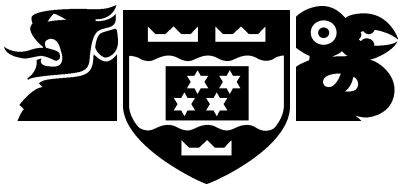
Pippin Barr, James Noble, and Robert Biddle

Technical Report CS-TR-02/20
September 2002

School of Mathematical and Computing Sciences
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Icons R Icons: User interface icons, metaphor and metonymy

Pippin Barr, James Noble, and Robert Biddle

Technical Report CS-TR-02/20
September 2002

Abstract

Icons are used in almost every graphical user-interface to computer software. Despite this, there is a serious lack of comprehension of what they are and how they work. The application of semiotics to user-interface icons can help to solve this problem by providing a framework for understanding icons in. This paper suggests such a semiotic approach and then applies it to real-world icons to show its effectiveness. Some hypothetical guidelines are derived from the application and a strong analytic technique results.

Publishing Information

A later version of this paper was presented at AUIC2003, the Australasian User Interface Conference, Adelaide, Australia, February 2003.

Author Information

Pippin Barr was a successful MSc candidate in Computer Science at Victoria University of Wellington; Robert Biddle and James Noble are members of the academic staff in the School of Mathematical and Computing Sciences and the School of Information Management at Victoria University of Wellington.

Icons R Icons

Pippin Barr **James Noble** **Robert Biddle**

School of Mathematical and Computer Sciences
Victoria University of Wellington,
Wellington, New Zealand
chikken@mcs.vuw.ac.nz kjx@mcs.vuw.ac.nz robert@mcs.vuw.ac.nz

Abstract

Icons are used in almost every graphical user-interface to computer software. Despite this, there is a serious lack of comprehension of what they are and how they work. The application of semiotics to user-interface icons can help to solve this problem by providing a framework for understanding icons in. This paper suggests such a semiotic approach and then applies it to real-world icons to show its effectiveness. Some hypothetical guidelines are derived from the application and a strong analytic technique results.

Keywords: semiotics, icons, usability

1 Introduction

Icons are one of the most popular graphical devices featured in graphical user-interfaces. Most common software packages such as word-processors, web-browsers, and file-system interfaces use them to convey their key functionality in a quickly comprehensible form. Given this wide use in industry, it is clear that graphical icons are an established feature of most interfaces that today's users encounter.

Despite this popularity, the availability of methods for analysing and evaluating computer icons is limited. The chief focus in icon research has been empirical studies of their effectiveness on groups of users, and largely common sense-based approaches to their design.

Semiotics is the study of signs. This paper proposes that we can gain significant insight from applying the methods of semiotics to computer icons. We will first outline the current situation of icons in more detail in section 2, then proceed to show how a semiotic approach can be created in section 3. Section 4 discuss how to apply this approach to the analysis of icons, after which we present some real applications in section 5. Finally, we discuss the use of semiotics in more depth in section 6 before pointing towards some related work in the area in section 7.

2 Background

It is a challenge to walk past any computer screen without encountering icons. They have become one of the most ubiquitous features of graphical user-interfaces for software of all kinds. Since their very early incorporation in the Xerox Star system, icons have seen increasing usage in computer software. Icons are used to indicate the core functionality of web-browsers, word-processors, the 'desktop' and many other programs.

The general consensus in industry is that icons are a key element to making user-interfaces usable. This is so much the case that using icons has become something of a default position, frequently used without any specific justification.

The use of computer icons is very strongly emphasised in various interface design guidelines. Apple's *Macintosh Human Interface Guidelines* has an entire chapter devoted to the topic. It suggests that "people often recognize pictures of things and understand them more quickly than they do verbal representations of the same things." (Apple Computer, Inc. Staff 1992) It is also offered that icons are a better choice for cross-cultural recognition than language is and, importantly, take up less space in the interface. Smith and Mosier's extensive interface guidelines include a suggestion as to the use of icons, recommending that they should "look like the objects or processes they represent" (Smith & Mosier 1986, Guideline 2.4/13). Bruce Tognazinni and Ben Scheinderman also offer guidelines for computer icon use in their books (Tognazzini 1992) and (Schneiderman 1992).

An icon is commonly defined as a small picture which represents some feature of the software it appears in. The feature is generally a system action or system object. When the icon represents an action it is largely intended to be interacted with to produce that action, for example when clicking on the 'print' icon in a word processor. When the icon represents a system object, the user may be able to act on the icon as a means to act on the system object, for example when double clicking on a 'document' icon to 'open' it. The important point here is that icons are intended to indicate the presence of some functionality or system object to the user.

Despite the overwhelming amount of use, and the many recommendations and guidelines available, there are surprisingly few principled methods for analysing exactly what icons do and are. Numerous studies have been performed seeking to assess whether icons are *useful* or not (Rogers 1986, Fullerton & Happ 1993, Magyar 1990), but analysis of the concept itself is rare. Instead, most designers have had to rely on a purely common sense interpretation.

Good design is, however, aided by a strong understanding of the concepts underlying the techniques used. In the next section we will provide a framework for better understanding icons through the use of semiotics.

3 The Semiotics of User-Interface Icons

Semiotics is the study of signs (Chandler 2001). In particular, it concerns *how* signs obtain their meanings and how they convey them. A sign is simply

Copyright ©2002, Australian Computer Society, Inc. This paper appeared at Fourth Australasian User Interface Conference (AUIC2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol. 18. Robert Biddle and Bruce Thomas, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

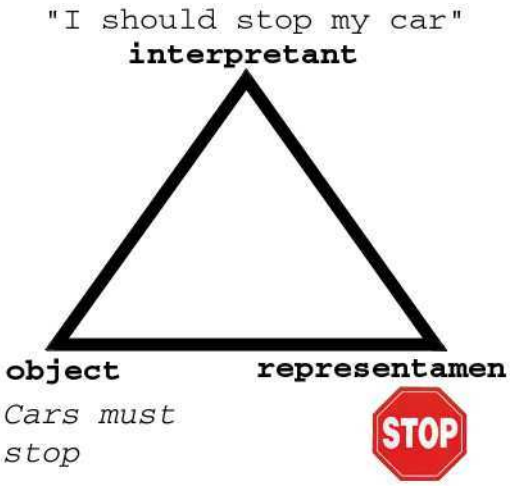


Figure 1: A Diagram of the Peircean Triad as Applied to a Stop-sign.

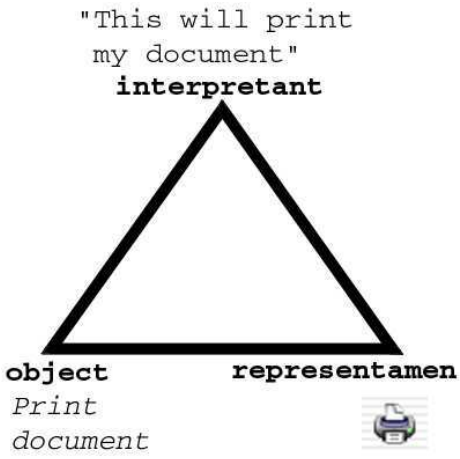


Figure 2: A Diagram of the Peircean Triad as Applied to a 'Print' icon.

Iconic Sign Indexical Sign Symbolic Sign

Figure 3: The three types of signs.

anything which stands for something else to some interpreter. With this broad definition, it is clear that almost anything can be a sign to someone. Thus, signs may range from words, to pictures, to footprints on a dirt track.

There are many approaches to the study of signs. For the purpose of this paper we will focus purely on the work of Charles Sanders Peirce. Peirce's semiotics is highly structured and taxonomic and is thus well suited to the task at hand.

Peirce proposed a triadic view of the sign. This triad is composed of the *object*, the *representamen* and the *interpretant*. We will consider the example of a stop-sign while explaining each of these. The object is the actual thing the sign represents or conveys: the necessity for cars to stop at a particular point in the road. The representamen is what does the representing: the stop-sign itself. The interpretant is the *result* of an interpreter's encounter with a sign: the thought that "I should stop the car." The whole sign can be represented in a diagram such as figure 1.

The basic relationship contained in the triad is that the *representamen* represents or conveys the *object* in a way that creates the *interpretant* in the mind of the interpreter. We are capable of purposefully *designing* signs to convey *intended* messages. In this case, the goal is for the representamen to effectively create an interpretant which matches the object. This is the case with the stop sign which was designed with the explicit purpose of creating the interpretant "I should stop my car" which matches the sign's object "cars must stop".

3.1 The Peircean Triad and Computer Icons

An icon in a graphical user-interface is another sign which is specifically *designed* with a purpose in mind. It is intended to convey some fact about the software in use to the user. Given that icons are signs, it is clear that we can apply semiotic analysis to them.

We can see that the three subdivisions of the sign can be straightforwardly applied to icons. Figure 2 demonstrates the application of the triad to an icon for printing a document in a word processor. In this case the object is the functionality available to the user: the ability to print the current document. The representamen corresponds to the actual graphic on the screen, the thing which represents the functionality to the user. Finally, the interpretant is the thought instilled in the user's mind when they see the icon or representamen.

3.2 Semiosis and Relations

Signs institute a process called *semiosis* in which the three parts interact to form a sign. Because icons are designed to convey a specific meaning, this process can be said to succeed or fail. The purpose of a designed sign is the conveyance of the object to the user as an interpretant. This success or failure is the relation between the object and the interpretant in the diagram. The degree to which the relation exists depends on how close the interpretant in the user's mind is to the object of the icon. We can call this the *effect* relation. It is defined by next two relations, and cannot exist independently of them. Conversely, the effect relation may *not* be present, while the other two are, if an icon is unsuccessful. This is when the user gets the 'wrong idea' and there is no relation between the interpretant and the object.

The success of the icon is defined by the above, but the process is effected *via* the representamen. Therefore the next two relations are those that actually *cause* the success or failure of the icon. The relationship between object and representamen describes *how* the representamen represents the object. In other words, it concerns the process of designing the icon to convey the meaning of the sign. We can call this the *representation* relation.

Completing the triangle is the relation between representamen and interpretant. This characterises the user's interaction with and reaction to the representamen. It is the process of the user forming an interpretant, and thus can be called the *interpretation* relation.

3.3 Icons Types

Peirce also created many classifications of sign *types*. Three types of signs identified by him are particularly interesting to us: iconic signs, indexical signs, and symbolic signs. These are all related in that each type is individuated by the representamen's relation to the object. In terms of computer icons, this means that the sign types are divided according to how the

icon relates to the underlying functionality. Figure 3 shows us the three sign types along with an example of each from a user-interface. An *iconic* sign [1] occurs when the representamen relates to the object through the process of *resemblance*. The icon for a document on the desktop is an image of a piece of paper which specifically resembles the object in question. An *indexical* sign occurs when the representamen relates to the object through causation. The 'print' icon in Microsoft Word is an indexical sign because the image of a printer represents the *cause* of the desired action of printing. Finally, in a *symbolic* sign the representamen relates to the object purely through convention. The image for 'reload', along with its textual caption, in the Mozilla browser is an example of this. There is no reason beyond common acceptance for that symbol or that text to signify the concept of 'reload'.

Each icon type has certain strengths and weaknesses. Examining these will provide a good basis for a simple evaluation of the effectiveness of icons without user-testing. We will demonstrate this process in section 5.

The chief benefit of both iconic and indexical signs rests in their being related to our normal perception of the world. In particular, they have the potential to be understood with minimal training of the user. In theory, a properly designed icon could allow a user to understand the meaning at the first encounter. Additionally, these icons have the ability to cross cultural boundaries: all humans have effectively the same physical perceptual system and so all will perceive the icons in a similar fashion.

What the icon actually depicts is, of course, another matter. The chief challenge with both iconic and indexical signs is choosing the representamen which will make the connection to the object most clear. In particular, the user must be able to *perceive* the connection between representamen and object. Note that icons also often involve the use of user-interface metaphors. Choosing good metaphors is a very large area of research in itself and will not be addressed here, instead see references such as (Wozny 1989, Carroll, Mack & Kellogg 1988, Erickson 1990).

The arbitrary nature of symbolic signs serves both as their advantage, and as their weakness. Because there is no perceptual connection between the representamen and the object, the sign must be learned by the user. Possibilities here include that the user is *shown* how to use the icon, is *told* via associated text such as a 'tool-tip' or *experiments* with the icon to find its use. Symbolic signs do not transfer well between cultures, as each culture tends to have its own set of symbols. The chief advantage of symbolic signs is that they are far easier to design, because they are arbitrary and thus lack constraints. Additionally, symbolic signs are easier to standardise, because they function through convention from the outset.

The divisions of sign types for icons here are not absolute. While symbolic signs are *purely* matters of convention, it is also true that iconic and indexical signs are conventional. This is because it is impossible to create an exact resemblance in the space available for an icon. The 'print' icon does not look *exactly* like a printer, for instance, but has enough of the conventional features that we recognise it anyway. Additionally, note that the print icon is an indexical sign, so recognising the image is a printer is not quite enough. We must also recognise the convention that an image of a printer refers to printing a document. Thus, all icons are symbolic or conventional to an extent.

[1] Note that where we use the word 'icon' we mean a computer icon. We will always use the term 'iconic sign' for the Peircean concept.

4 Icon Evaluation

The process of icon evaluation amounts to considering whether the interpretant that results from the user's encounter of the representamen is of the right kind. This result, which we called the *effect* relation earlier, comes out of both the *representation* and *interpretation* relations. The preceding discussion of sign types has given considerable attention to the representation relation. Note, however, that the interpretation relation is a *subjective* matter. The interpretant is created entirely by the user and is created in the context of that user. This makes it difficult to create principled means of predicting the relation.

Despite this, the semiotic approach does in fact provide valuable guidance as to the entire process of icon understanding. While the focus is on the representation relation, we can also provide insight into the interpretation process. In particular we can analyse the suitability of the icon *type* to the task it is designed to serve. Additionally, we can use the terminology provided to look at icons in a more detailed fashion when applying traditional common sense analysis. The triad should reveal more interesting insight because of the way it gives a structure to discussion.

Icons almost always exist in the context of other icons. More importantly, they exist within the context of the functionality of a particular piece of software. Because of this, the interpretation relation is not quite as free as one might imagine. Specifically, the user will generally know the *kinds* of functions that should be available. This knowledge will affect their interpretation of various icons, because they are able to dismiss highly unlikely explanations and focus on relevant ones.

An icon's relation to other icons may often reveal its purpose as a function of its difference or similarity to the others. Note also that icons often appear with associated text or 'tool-tips' that aid the user in the process of interpretation.

Finally, we ought to point out that icons are not necessarily meant to cause the user to *recognise* the functionality immediately, but can be used to help them to *remember* it. This is one reason why we might prefer indexical and iconic signs: they aid the user's memory of a aspect of the software through a resemblance or causal connection.

5 Case Studies

We will now outline three case-studies which use the semiotic approach discussed to analyse particular sets of computer icons.

5.1 Library Widgets

Figure 4 shows a selection of icons from a proposed standard icons set for bibliographic databases (*Bibliographic Database Applications Standard Icon Set* 2002). The icons are divided into three categories: *actions*, which are things you can do in the system; *fields*, which represent aspects of records; and *operators*, for performing searches. We have included three icons from each group in the figure.

The 'clear' icon is an indexical sign: a 'trash-can'. Although this *is* a standard icon used in interfaces, note that it is traditionally used as sign for *file-deletion* on the desktop. Using it for clearing a form here will be confusing to users. This is particularly because the 'trash-can' is an *iconic* sign in the desktop, but an *indexical* sign here. On the desktop the trash-can represents a system object to be interacted with, whereas the library icon refers to the

Figure 4: A selection of icons for a library database.

function of data-deletion. Note also that the traditionally association of *deletion* with the trash-can is much stronger than the notion of 'clearing' a form. The other two actions are also indexical signs. The icon for 'going home' is an image of a house, and the icon for using the thesaurus is an image of an open book. Both images reflect what the *result* of the action taken by the icon is. Note that the thesaurus icon is somewhat vague and we can imagine a range of possible interpretants. In particular, because *books* are the staple of libraries, it might well be difficult to interpret the image of a book as being *specifically* a thesaurus. Note also that we might consider the image of the book to be *iconic*, indicating a 'book' system object. This also applies to the 'clear' icon, which might be thought of as referring to an actual 'trash-can' object in the system. There is a serious issue here.

In the icons for 'fields' we find instances of each icon type in the three examples available here. The author icon is an indexical sign, because it shows the *result* or *effect* of an author, which is writing. At the same time, we can consider it to be an iconic sign because it *resembles* the author through their hand. Note that icon *context* is important here, because it ought to prevent the user from thinking that it might be their *own* hand writing and hence some kind of editing function. The subject icon is largely symbolic. It may also be iconic, resembling the marking piece from the game *Trivial Pursuit*, or a pie-chart. Given that this seems to be the only instance of such metaphors, it will likely be confusing to users. That said, it may be acceptable as a remembering device. Finally, the keyword icon is quite strange. There seems to be an attempt to create an iconic sign, by having the icon *resemble* a key and hence create a kind of visual pun. Given the somewhat tenuous nature of this relation, though, it seems that the icon is also a symbolic sign. Finally, note that if the user expects iconic signs for system objects, there is a likelihood they will interpret the key as relating to a *security* metaphor, rather than to graphical word play.

All of the icons in the operators category are symbolic signs. Two of the examples come from mathematics in the greater-than and not-equal symbols. The final icon uses the actual *word*-symbol from the English language to represent the operator. These icons therefore require a knowledge of mathematics and also logic in order for the interpretation process to be successful. Given that this is quite common knowledge, the assumption seems fairly safe. This is an instance in which the use of highly conventionalised symbolic signs is well advised. Note that it is only possible, however, when the object of each sign matches an accepted use of the representamen.

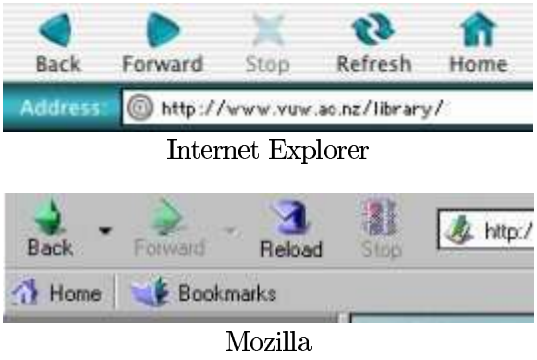


Figure 5: Examples of icons for identical purposes from two different browsers.

Summary

The suggestions of confusion over the key, author and thesaurus icons emphasise the value of the semiotic approach. It has been possible to analyse *which* icons might be problematic, as well as what problems might arise, which lends a focus to the later user-testing process.

The icons in the *fields* section were the most problematic set. This was reflected by their uncertain categorisation in the sign-types discussed. The blurring between sign-types means that users will likely be uncertain as to how to exactly interpret the icons. Thus, confusion may well arise on encounter with them.

Finally, we saw that appropriate use of symbolic signs can be very successful, provided that the object for the sign matches a traditional object used with the representamen, and thus is likely to lead to the correct and conventional interpretant.

5.2 Explorer Icons versus Mozilla Icons

We can also use the semiotic approach to compare two sets of icons for the same functions. For example, figure 5 displays the navigation tool-bars for two popular browsers, Internet Explorer and Mozilla. The common functionality involves moving back and forward through web-pages, stopping transfers, reloading pages, and returning to the user's home-page.

The icons for navigating back and forth between pages are highly standardised in the browser industry and consist of left and right arrows. Clearly these are symbolic signs, which is appropriate, given the lack of an underlying metaphor for this process (although it is possible that the rewind and fast-forward buttons of a tape-deck are the inspiration). The 'refresh' or 'reload' signs for both are also symbolic signs. In this case, however, there are two different signs used. Both have some semi-conventional representation of a 'cycle' concept, but other than this they rely on the user learning the convention.

Both browsers use a small image of a house for the function of 'going home'. Internet Explorer's icon is considerably less photo-realistic, however, and has thus moved towards being a symbolic sign. This has apparently been done to fit into an overall design style for the browser's tool-bar. The Mozilla icon is far

Adobe Acrobat Sherlock Library standard

Mozilla Apple Mail

Figure 6: Examples of icons for searching or finding from different applications.

more representational, and is thus firmly in the indexical category: the result of the operation of 'going home'. Again, however, *both* of these icons are quite conventional within computer navigation systems, and so *both* are also symbolic signs.

Possibly the most interesting of the icons in the tool-bars are the icons for stopping a transfer. The icon used by Internet Explorer is purely symbolic, thus fitting in with the overall symbolic design of the tool-bar. The 'X' symbol is commonly used by people to indicate cancellation or negation. Note that sign must be interpreted in its context to conclude exactly *what* this cancellation or negation is applied to. Given that the surrounding icons all deal with the loading of web-pages, there is a good chance users will understand this symbolic sign. Additionally, the icon is only 'available' when a transfer is taking place, which aids the contextual interpretation. The possibility for misinterpretation is still present of course.

The Mozilla stop icon is an indexical sign. It represents a traffic-light displaying a red light. Interestingly this exhibits multiple levels of signs. The icon itself is an indexical sign, representing a real-world *cause* of stopping. A traffic-light is, in itself a *symbolic* sign, representing the object of 'stop (your car)' through a particular colour. In fact, if we are to be precise, the traffic-light exhibits what Peirce would call a qualisign, a sign made up of a quality (in this case the quality of 'redness'). Again note that context is required to realise that the icon does not refer to stopping *cars* but instead to stopping *transfers* of data. Also, some confusion might arise because traffic lights traditionally tell *us* to stop, and are not used by us to stop other things.

Summary

Overall we can see that most of the icons utilised by the two browsers are symbolic signs. This is likely because there is no dominant metaphor for the internet, and thus no real-world phenomenon to create iconic and indexical icons from. Netscape established a fairly firm standard early on, however, so the abstraction nature of browser icons is not overly problematic.

The stop icons are an interesting example of design. Mozilla uses what we might call an indexical qualisign, attempting to use at least some metaphor in their tool-bar. Internet Explorer on the other hand, has chosen to make their icons completely or largely symbolic, perhaps to give the browser a particular style.

5.3 Search Icons

Functionality for performing searches is present in a very large number of applications. Because searching is so popular, it is often represented iconically on a tool-bar of some kind. In this section we examine search or find icons from five different applications displayed in figure 6 and consider them from the semiotic standpoint.

First of all, we can categorise these icons via the imagery used in the form of the representamen. Two use a magnifying glass in some way, two use binoculars, and one uses a flash-light. If all of these search functions were for identical purposes this might be of concern because the various images are so disparate. The applications are all different, however, so this should not be a problem. This does call to mind the importance of consistency, though.

A magnifying glass is a *tool* used when searching for something *small*, generally among a lot of other small things. The Apple Mail icon is used in searching for emails among a collection of emails, while Sherlock is an application for searching a computer and the web. Note that the Sherlock icon also suggests 'Sherlock Holmes' through the use of a deerstalker hat. Both Holmes and the deerstalker are strongly cultural references, and will not be understood by everyone. Note also that the chief function of a magnifying glass is to visually *enlarge* something. There is therefore the danger of interpretants that suggest the icon represents a magnification tool. Finally, magnifying glasses are traditionally used for examining within documents. Because of this, the Sherlock function, which actually searches the entire computer, might seem slightly out of line.

A magnifying glass can be considered as an iconic sign to some degree. It resembles the tool used when searching. It can more strongly be considered as an indexical sign because the tool is involved in the causal process of searching for something.

Binoculars are used to search for something *far away* from us and, again, to *enlarge* a particular portion of the visual field. The Adobe Acrobat icon is for searching the current document, while the library standard icon concerns finding a record in a database. In both cases the implications of the representamen seem slightly wrong. They suggest that the things we are looking for are very distant from us and we wish to see them closer up. Binoculars also suggest a process of scanning, attempting to find the thing we are looking for, rather than a systematic approach.

Like the magnifying glass, the binoculars are a tool used in the search process. Hence, as above, they seem to be somewhat iconic but largely indexical. While they do seem to represent the action of searching well enough, it is questionable whether the implications of binoculars fit the actual task well. It could well be that this is a response to the worry mentioned above that a magnifying glass might suggest a magnification tool, rather than a search tool. This is especially true in the context of Acrobat Reader, where enlargement of the document is a legitimate function.

A flashlight is used to search dark areas and to find objects hidden in the darkness. They are good when we cannot *see* the area we are searching, and illuminate only particular parts of it. This fits well with the concept of searching the internet, which is almost completely unseen at any given time. A search engine reveals those parts of the internet we are interested in, while the rest remains hidden. The flashlight is again largely an indexical sign, relating to the causal process of searching for something.

Summary

Although all of the icons discussed above are indexical signs, iconic interpretations of each are possible. A user might think that there is, for example, a *binoculars* system object as part of the software. Because

this does not fit well in the context of any of the applications, though, it will likely be dismissed. Note, finally, that all of the indexical signs discussed concern a search in a *metaphorical* world, rather than the actual searching that takes place in a computer, so there is a level of indirection here.

6 Discussion

Having now shown that the approach outlined in this paper works when applied to real user-interface icons, we can have a large degree of confidence that the process is a useful one. At this point we will consider some issues with the approach, as well as outline some interesting implications that follow from it.

One point of criticism that could be leveled at the above approach to icon evaluation is the lack of a formal way of judging the expressiveness of icons. The complaint might go that the method is handicapped by its inability to rigorously predict what sort of reaction various icons will get.

While this would be nice, it is an idealistic wish for icon evaluation. It is equivalent to formalising the interpretation relation we discussed earlier. As we noted then, it is difficult to formalise a means of evaluating how real users will react to *anything*, and this applies just as much to icons. The contribution of the approach in this paper is that we have uncovered another relation, the *representation* relation, which we *can* talk about in a principled manner. Thus, we have provided a solid analysis of *half* of the process of creating a good icon. In fact, it is the only half that *can* be addressed in a strict fashion.

In addition to this, we have seen that the semiotic approach provides a good structure for the common sense evaluation of icons. We have shown in the case studies that using semiotics gives considerable guidance as to predicting what will be problematic, for example. It seems clear, then, that the approach outlined in this paper should be used in *conjunction* with user-testing to gain the best results. This is particularly true of sign-type classification, which yields important insight into the likely success of an icon. We can, in fact, provide some postulate guidelines, based on this.

> *Icons for representing qualities or system objects should be iconic signs.*

Iconic signs, because they involve the relationship of *resemblance* are best at representing system objects and qualities. In this case the representamen needs to call to the user's mind the underlying object or quality that the sign represents. An example of this is the document icon common to the desktop. The icon *resembles* the system concept of a document, and so is an iconic sign.

> *Icons for representing system functionality should be indexical signs.*

Indexical signs, because of the relation of *causation*, are best for representing *actions*. We can create sub-categories here, also. The icon represents either the effect or cause of a particular action, and this in turn should create an interpretant which considers the *action* itself. The printer on a 'print' icon represents the *cause* of printing, while the blank sheet of paper on a 'new document' icon represents its *effect*.

> *Icons for representing non-metaphoric concepts in the system should be symbolic signs.*

It should be clear that symbolic signs can effectively convey any type of information, due to their arbitrary nature. In other words, because there is no direct connection between representamen and object, there is no need for the representamen to look any particular way. The connection is simply learned.

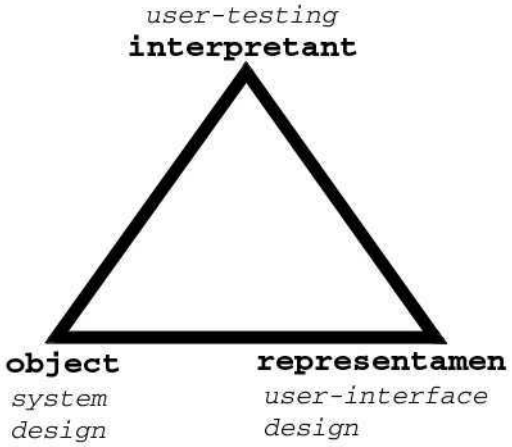


Figure 7: A triadic view of the software design process

We have seen that confusion between sign-types can lead to confusion of the purpose of an icon and misinterpretation. The typical instance of this confusion is between the iconic and indexical types. They both represent objects, the difference being the role of the object in the representation. Fortunately, convention and the context of the icon will generally solve this particular issue.

Having terminology in place as well as a model of the process of icon signification is useful for helping designers understand how their work will be received by users. It enables better discussion of designs, and helps to focus attention on particularly important parts of design. One especially useful aspect of using the triadic view of the icon is its equal emphasis on the three parts of the sign. In particular it shows that we must be aware of all aspects. Often designers might become most focussed on the representation relation, and thus lose sight of the importance of the *interpretant* and thus, effectively, the importance of the user. While the triad does not give specific guidance on attending to the interpretation relation, it does make its important and presence clear. This reinforces such usability guidelines as Jakob Nielsen's mandate to "know the user" (Nielsen 1993).

The triad can also be super-imposed on the *people* involved with any piece of software. The *object* relates to the the actual system designer, who creates the "meaning" of the software. The *representamen* matches with the user-interface designer. Finally, the user is identified with the *interpretant*. In terms of artifacts of the software process, we can consider the code or specification of the system as the object, the user-interface itself as the representamen, and the user's response and interaction with the system as the interpretant.

This, in turn, relates to Donald Norman's triadic view of interaction models (Norman 1988). In Norman's theory, there are three models: the system, design and user models. Each of these relates quite simply to the Peircean triad, with the system model being the object, the design model being the representamen and the user model being the interpretant. The purpose of an interface can be seen as the attempt synchronise these three models such that the user will understand what the system does. This is effectively a version of semiosis.

In fact, the triad can be generalised to represent the entire design process of software. The triangle

emphasis the importance of each relation equally, and suggests that design should take place with consideration of each aspect all the time. This is in opposition to what might be considered the typical process of: building the software or *object*; putting an interface or *representamen* on it; and then finally performing user-testing, or addressing the *interpretant*. This is a linear process, as opposed the unified view offered by the triad. We can represent the unified view in a simple manner, as in figure 7. This also emphasises that the triadic view can certainly be generalised to the analysis of user-interface design in *general*, despite our limiting the focus of this paper to icons.

7 Related Work

There has been a significant amount of work done applying the theory of semiotics to computers in various ways. Joseph Gougen has developed a theory of algebraic semiotics, through which he hopes to formalise the process of user-interface design (Malcolm & Goguen 1998, Gougen 1999). Peter Bøgh Andersen has done considerable work on his theory of computer semiotics, where he examines the application of semiotics to the entire computer system (Andersen 1997, Andersen 1992). He has particularly focussed on interface design and programming as a sign-system. A short paper by Bøgh Andersen analyses how far the use of semiotics in computer human-computer interaction can take us, concluding that perhaps some hopes are pinned too high (Andersen 2000). His basic ideas are that semiotics is useful for transferring insight from other media, and as a framework for assisting in practical design. Kevin Mullet and Darrell Sano provide a very clear analysis of the application of semiotics to user-interface design in (Mullet & Sano 1995), including some consideration of the application of the sign-types. The work of Mihai Nadin is also a very good example of applying semiotics to human-computer interaction (Nadin 1988). Jean-Marc Orliaguet has provided an extremely detailed Peircean look at interfaces with the objective of 'programming' the user's behaviour as far as possible (Orliaguet 2000). This is effectively the attempt to control the interpretation relation as far as possible. Although the conclusion is that the issue has not really been solved, a significant, if complex, assessment of the applicability of Peirce to interfaces is provided. James Noble and Robert Biddle apply semiotics to the software engineering concept of design patterns in (Noble & Biddle 2002), and Marcelo Pimenta and Richard Faust offer a semiotic approach to requirements gathering in (Pimenta & Faust 1997). These papers show how semiotics appears to be highly applicable to the entire design process as mentioned in section 6. Colin Ware's categorisation of signs into *sensory* and *arbitrary* is also worth consideration when discussing sign-types (Ware 2000).

On the side of icon evaluation and design, there is a very large body of work. The large part of research involves empirical tests of particular icon sets to examine their effectiveness (Rogers 1986, Fullerton & Happ 1993, Magyar 1990). This work helps us to understand the interpretation relation and offers methodologies for testing it. Work on formalising the design or analysis process is somewhat more scarce. There have, however, been entire books on the process (Horton 1994) as well as detailed papers (Gittins 1986).

Despite the body of work available, there has been little or no attempt to directly focus the use of semiotics on user-interface icons. The works on semiotics referred to above do occasionally mention the topic, but a detailed analysis is never given. The research into icon evaluation and design is heavily focussed on empirical studies of icon effectiveness, and common sense guidelines. It is apparent, then, that the present work is original in its strict application of a semiotic framework to icon analysis and evaluation.

8 Conclusion

In this paper we have explained how some of Charles Peirce's theory of semiotics can be used to better understand the design and evaluation of computer icons. We outlined how the Peircean triad and three of Peirce's categories of signs are all applicable to the icons we encounter every day in our computer systems. We have also drawn connections to other areas. We showed, for example, that the triad has links to other triadic views of the software process, and ultimately has a bearing on software design in general.

Through our case-studies we showed that the semiotic approach can yield useful insights into how icons work. We then offered further analysis of how semiotics is useful and showed that the approach in this paper is a novel one.

References

Andersen, P. B. (1992), 'Computer semiotics', *Scandinavian Journal of Information Systems* **4**, 3–30.

Andersen, P. B. (1997), *A Theory of Computer Semiotics*, 2nd edn, Cambridge University Press.

Andersen, P. B. (2000), 'What semiotics can and cannot do for HCI', in CHI'2000 Workshop on Semiotic Approaches to User Interface Design.

Apple Computer, Inc. Staff (1992), *Macintosh Human Interface Guidelines*, Addison-Wesley Publishing Company.

Bibliographic Database Applications Standard Icon Set (2002), http://www.scran.ac.uk/cgi-bin/IFLA_Icons.pl.

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.

Chandler, D. (2001), *Semiotics: the Basics*, Routledge.

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.

Fullerton, S. & Happ, A. J. (1993), A user-oriented test of icons in an educational software product, *in* 'Proceedings of the Fifth International Conference on Human-Computer Interaction', Vol. 2, pp. 44–49.

Gittins, D. (1986), 'Icon-based human-computer interaction', *International Journal of Man-Machine Studies* **24**(6), 519–543.

Gougen, J. (1999), An introduction to algebraic semotics, with applications to user interface design, *in* C. Nehaniv, ed., 'Computation for Metaphor, Analogy and Agents', Vol. 1562 of *LNAI*, Springer-Verlag, pp. 242–291.

Horton, W. (1994), *The Icon Book: Visual Symbols for Computer Systems and Documentation*, Wiley.

Magyar, R. L. (1990), Assessing icon appropriateness and icon discriminability with a paired-comparison testing procedure, *in* 'Proceedings of the Human Factors Society 34th Annual Meeting', Vol. 2, pp. 1204–1208.

Malcolm, G. & Goguen, J. A. (1998), Signs and representations: Semiotics for user interface design, *in* R. Paton & I. Nielson, eds, 'Visual Representations and Interpretations', Springer, pp. 163–172.

Mullet, K. & Sano, D. (1995), *Designing Visual Interfaces*, Prentice Hall.

Nadin, M. (1988), 'Interface design: A semiotic paradigm', *Semiotica* **69**(3), 269–302.

Nielsen, J. (1993), *Usability Engineering*, Academic Press.

Noble, J. & Biddle, R. (2002), Patterns as signs, *in* B. Magnusson, ed., '16th European Conference on Object-Oriented Programming', pp. 368–391.

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

Orliaguet, J.-M. (2000), 'How do we reason when using computers? how programmable are we?', http://www.medialab.chalmers.se/people/jmo /essays/how_do_we_reason.pdf.

Pimenta, M. S. & Faust, R. (1997), 'HCI and requirements engineering - eliciting interactive systems requirements in a language-centred user-designer collaboration: A semiotic approach', *SIGCHI Bulletin* **29**(1).

Rogers, Y. (1986), Evaluating the meaningfulness of icon sets to represent command, *in* 'Proceedings of the HCI'86 Conference on People and Computers II', pp. 586–603.

Schneiderman, B. (1992), *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd edn, Addison-Wesley Publishing.

Smith, S. L. & Mosier, J. N. (1986), 'Guidelines for designing user-interface software', ESD-TR-86-278. Bedford, MA 01730: The MITRE Corporation.

Tognazzini, B. (1992), *Tog on Interface*, Addison Wesley.

Ware, C. (2000), *Information Visualization: Perception for Design*, Academic Press.

Wozny, L. A. (1989), 'The application of metaphor, analogy, and conceptual models in computer systems', *Interacting with Computers* **1**(3), 273–283.