User Interface Metaphors

Rainer Brockerhoff

***Abstract**: Cognitive science is taking a revolutionary turn, away from abstract theories of the mind and towards a recognition of the essentially embodied nature of human thought, processing concepts in terms of conceptual metaphors. Computer user interfaces, as complex metaphors in their own right, can now be discussed in terms of their relationship to our basic cognitive processes. As a rule, user interface metaphors that are solidly grounded in physical concepts and experiences may be considered more "intuitive".*

A Brief Introduction, Interspersed with Warnings, Disclaimers, Apologies, and Miscellaneous Weaseling

Up to a few decades ago, western philosophers and scientists have been working from a foundation which has usually been considered too obvious to state explicitly; namely, that reason is an abstract, disembodied, universal and purely formal entity. This underlying assumption is so pervasive that I at first wrote "philosophy and science" instead of "philosophers and scientists"!

A recent revolution in cognitive science has pulled the carpet out from under this assumption. I hasten to add that my academic qualifications are insufficient to do more than give a brief summation of my understanding of the scholarly works cited at the end of this paper. Furthermore, many sectors of the scientific and philosophical establishments appear to remain unconvinced of the arguments exposed therein.

My main reference here is "Philosophy in the Flesh" [1], by George Lakoff and Mark Johnson, which I will irreverently refer to as "**PitF**". Any inaccuracies in transcription and interpretation are my own.

The classical assumptions state categorically that:

- We can know our own minds by introspection,
- Most of our thinking about the world is literal, and
- Reason is disembodied and universal,

However, **PitF** demonstrates empirically that:

- Most thought is unconscious. What is available to introspection is a very small portion of the mind's processes.
- Abstract concepts are mostly metaphorical. Basic metaphors derived from bodily conditions and experiences are necessary to form even a simple thought.
- Mind is embodied. The structure of our thought requires a body.

You may question the use of the word "metaphor", since it is a common misconception that this refers only to literary analogies; however, contemporary linguists have considerably enlarged the meaning of "metaphor" [2,3,4,5]. For our purposes, we can consider a metaphor to be a "mapping" function that projects the structure of some source domain — usually a basic bodily perception or movement — onto a target domain (see the Lakoff & Nuñez paper for a more complete explanation).

Grounding metaphors map common physical experiences onto a target domain; examples would be "movements are changes" (exemplified by the phrase "we *came* to an agreement"), and "time is money" ("he *invested* much time on this paper").

There are also *complex* metaphors, which are higher order mappings of simpler metaphors, used to express complex ideas, like "the mind is a machine" ("he had a *mental breakdown*"), and *linking* metaphors which map different domains onto each other, like "the number line", which maps arithmetic onto geometry. Both of these are used to extending our metaphorical reach into regions that were not accessible to thought beforehand.

It must be stressed that all these metaphors operate effortlessly and below the level of conscious awareness. The definitions and examples are simply surface manifestations of metaphorical thought, the tip of the tip of the iceberg of the mind. Any new metaphor we make up consciously *must* use the mechanisms of our unconscious, everyday metaphor system; however, it seems that the ease of applying the new metaphor depends enormously on the number and complexity of levels separating it from the grounding metaphors.

PitF, after a detailed explanation of these and other basic concepts of what the authors call "Second-generation cognitive science", goes on to an analysis of the underlying metaphors of classical and modern philosophical theories, with often surprising and profound results. I also refer the interested reader to other publications which apply this methodology to mathematics [6], arithmetic [7], politics [8] and […].

Hopefully you are now suitably cautioned in taking the following *cum grano salis* regarding my interpretation of the general theory, and sufficiently interested in considering how all this might be applied to actual computer interfaces.

Finally Sort Of Getting to the Subject, and Some Historical Considerations

To get myself back on more familiar ground, I'll quote from John Lawler's 1987 lecture "Metaphors We Compute By" [9]. In the section "The Desktop: the Computer is a Workplace", he says:

> We're all familiar with the Macintosh Desktop and its origins in the Xerox Star and its copies in Windows, etc. And we've all had lots of discussions about how great an advance it is in user interface design (whether we believe that or not, there are enough folks who do to involve us in such discussions almost endlessly). You have to admire that kind of enthusiasm, and the products that evoke it. Nevertheless, we've not yet arrived at the perfect user interface.
>
> … I don't really see a great deal that can be done about it, in fact, beyond making user interfaces as customizable and flexible as possible, and using a lot of synonyms when designing them. The point I want to make here is that diversity in personal styles of information management is not yet a well-known or -handled part of user interface design. There's always a big problem with adaptation; either you have to adapt yourself to the design of the computer (and you may not be able to do so usefully), or you have to adapt the computer to your own strategies (and this is a very difficult task at best). Mostly we try to do both, with quite variable degrees of success.

At the time of Lawler's lecture — he was an early Mac enthusiast — the idea that metaphors were useful for human-computer interaction was already well-established in the minds of Apple's human interface designers, and they of course were based on the groundbreaking concepts developed at Xerox PARC, and other research centers. Even so, this appears to be the earliest academic reference to human-interface metaphors outside of the computing field.

Before the appearance of the Graphical User Interface (GUI) and the invention of the mouse by Douglas Engelbart, the basic metaphor for human-computer interaction was what Lawler defines as "Computing is a linguistic activity". Not only were computer interfaces text-based and linear, but in both popular and technical views of the computer knowing a "computer language" was basic to interacting with a computer. From the 1950's to mid-1970's, operating and programming a computer were practically synonymous.

In the same time frame, Noam Chomsky's models of generational syntax and transformational grammar constituted the orthodoxy of linguistic thought. These ideas of, ultimately, reducing human language to abstract syntactic and semantic schemata ran parallel to developments in computer science, where more and more complicated "languages" were devised, and in artificial intelligence, where symbol manipulation and logical inference were seen as the obvious way to model the human mind. Philosophical reductionism seemed poised to conquer the world. Household robots and machine translation were considered to be "just around the corner"…

Alas, it was not to be. Soon Chomsky's disciples were scattered in wildly different directions in what are now termed the "Linguistics Wars" [10]. The two-level syntax fiasco put an end to the complexification of computer languages (although the recent ANSI C++ standard's complexity reminds me very uncomfortably of the "Algol 68 Report"). Artificial intelligence hit a conceptual brick wall; in my opinion, the wall has not been breached but only pushed further along by brute-force techniques. And, of course, household robots are still as far off as they were in the 1960's, and machine translations by BabelFish and other efforts are the source of much merriment and derision in the Internet community.

At the same time that cognitive science was making the leap from the "abstract reason" to the "embodied mind" model, computer interfaces went graphical. Suddenly, computer users were no longer required to learn an abstract command language to operate a computer. Icons, graphical representations, point-and-click and copy-and-paste became **the** paradigm of the computing experience. A parallel effort to apply these concepts to programming has been much less successful, at best resulting in better representation of high-level relationship in programming entities, at worst spawning a whole generation of Visual Basic professional amateurs.

The success of the GUI was accompanied by unsupported claims that such interfaces were somehow more "intuitive". My personal experience was that they were easily learned — at least by somebody already well versed in earlier technologies — but explanations about **why** they were intuitive were not completely convincing. And as anybody who tried to teach an elderly relative to use a word processor or e-mailer knows, they're not all that intuitive to laypersons.

Unfortunately after the comparatively sudden takeover of GUIs, no substantial refinements or new paradigms have appeared. Color, drag-and-drop, translucency, throbbing default buttons, and so forth are merely refinements but not revolutionary developments. Three-dimensional interfaces and "Virtual Reality" are still hampered by inadequate hardware and lack of theoretical underpinnings. And, of course, there still are many people who consider GUIs deficient in one way or another; for an extremely interesting and articulate argument, see Neal Stephenson's essay "In the Beginning Was the Command Line" [11].

Where's the Metaphor?

Fortunately, second-generation cognitive science can come to our rescue. So-called "abstract" thought processes are implemented by, to quote Lakoff and Nuñez, "image-schemas, basic-level concepts, idealized cognitive models, prototypes of various kinds, radial categories, conceptual metaphors and metonymies, mental spaces, and conceptual blends".

Let's try to see what grounding metaphors may apply to the current crop of graphical user interfaces, and specifically to the Mac OS. Much of what follows may seem obvious, but there are actually many instances where these metaphors are employed in conflicting or inconsistent ways. In other disciplines, conflicting metaphors are actually more common than one might expect, and are considered as enriching, rather than devaluing, the subject. However, when designing user interfaces, consistency is a powerful tool, since one usually expects the user to learn from solitary experience, rather than from interaction with others users.

The Computer is a Robot That Understands Only Robot Language.

This is the metaphor used for command-line user interfaces and for actual computer programming. Distant from physical reality by many levels of complex and linking metaphors, this is much harder to use and understand. The user has to adapt to keyboards, arbitrary commands, linear text displayed on a screen or printed out, an artificial language with none of the flexibility of natural languages, and so forth; all of these being very high-level metaphors themselves, and invented to ease the computer's tasks instead of the user's.

Both Stephenson and Lawler, as former Mac users, have argued that the Mac's complete lack of a command line unduly restricts a knowledgeable user. It will be interesting to hear their opinion of Mac OS X, when it arrives next year, as it will offer both a GUI and a command line. In the meantime let's concentrate, for the rest of this paper, on the metaphors underlying the GUI part.

The Computer Screen is a World.

This is the first and most obvious metaphor for GUIs. The user is put quite naturally in the position of a benevolent god looking down on a toy world, which he can manipulate at will. Then again, some would say, the user is more akin to the hapless owner of an aquarium in which the fish have run amok, are gobbling each other up, and are drawing obscure runes in the sand which he suspects are elaborate obscenities in fish language… but I digress.

The Mouse is a Hand (and Therefore, the Cursor is a Finger).

This too is a most obvious metaphor. Therefore, moving the cursor over an object on the screen is "pointing". This is usually done through the intermediation of a mouse, or "pointing device". Pointing does not imply "touching" the screen object; to do that, you need to first point and then "click" the mouse button. "Grasping" an object on the screen is "continuously touching" by holding the button down. "Dragging" is moving the mouse while holding the button.

Compared to typing on a keyboard, this is much nearer to physical reality. Since the actual implementation (the mouse) is both sensorially and functionally impoverished, there are feedback actions (object aspect changes, selections, audio feedback, cursor shapes, marquees and so forth) to complement the user's perceptions of what he's doing; and modifiers (double-clicking, multiple buttons, holding a key down, etc.) to extend the range of possible actions. These extensions have to be carefully thought out, so as to either match actual sensory or manipulatory conditions, be easily deduced from cultural context, or be easily memorized by a suitable mnemonic — the latter actually being an explicit metaphor used as a cognitive tool.

In particular, multi-button mice are intrinsically harder to use, as in real life all fingers are conceptually equivalent in terms of pointing, touching or grasping. Therefore, both multi-button clicking (and its Mac equivalent, option-, shift-, or command-clicking) are counterintuitive and have to be carefully documented. Then again, once one gets used to the necessary movements, they are easily retained in muscle memory.

Since mouse movements are usually mapped quite faithfully onto cursor movements, and finger movements for clicking are analogous to tapping on something, this is very easily learned (and therefore might be termed "intuitive" by some). Here too muscle memory" is very helpful, and indeed pointing and clicking uses the motor/sensor feedback apparatus we use for all our physical activities.

The Nature of Thought

Other metaphors do not use our bodily appendances in such a direct way. Indeed, most metaphors would usually be considered as having no physical equivalent at all — that is why abstract thought has been considered, for so long, to have an existence that is completely independent of the mind. However, **PitF** conjectures that when so-called "abstract thought" first appeared in primitive humans, they naturally used the same neuronal circuits that they used for executing the physical actions implied in the appropriate grounding metaphors. These circuits are suppressed from having external effects by the same mechanism that suppresses body movements during dreaming.

Abstract thought is thus exposed as a special kind of "lucid dreaming". This conjecture about the transition from proto-human to human seems to be quite reasonable, for the following reasons:

- Other animals are observed to display behavior implementing many of the simpler grounding metaphors, and some of the more complex ones — especially regarding relationships within a group. Therefore, the transition to human thought would be easier for a non-specialist species, which necessarily would have a more varied range of grounding metaphors in its neural arsenal.

- The motion suppression mechanism is already in place in many mammals — anybody who has seen a dog dreaming will understand what I mean — and therefore the transition to human thought demands only an adaptation of this mechanism, not a broad and revolutionary emergence of a completely new one, like other theories demand.

- Spontaneous gestures accompanying speech often trace out images from the source domains of conceptual metaphors. As the joke goes, when you kidnap an Italian, you don't gag him — you have to tie his hands so he can't call for help.

- Studies of language acquisition in infants show that the capacity for abstract thought is gained gradually and by continual linking and compounding of grounding metaphors.

- Anatomical corroboration comes from the observation of so-called

> "mirror neurons" in primates, as explained in V.S. Ramachandran's 1995 talk at the Society for Neuroscience[12]. These neurons are activated both when a monkey performs a certain action, and when he observes another monkey performing the same action — a sign that the action is being processed conceptually, rather than on the purely motor or sensorial level.

Icons are Objects (and similarly, words, ranges of text, windows and other graphical entities are objects).

This simply allows the mouse metaphor to work properly — since there's no use in having a metaphorical hand without things for it to touch, or grasp, or point at. For this to be effective (or "intuitive"), screen objects have to be consistent — not only should any object that *looks* clickable (or draggable, etc.) actually *be* clickable, but the feedback for any action should be consistent with the action's context, and similar objects should act similarly.

Of course, objects which can be "handled" by the user should be suitably distinguishable from "decorative" objects, to avoid wasting the user's time with fruitless exploration. Style is a major factor in making objects easy to use, and perhaps the hardest part to teach to an interface designer. Here we also have sensory restrictions, since we're dealing with images on a 2-D screen, not with the 3-D objects (complemented by sound, smell, taste and touch impressions) that we're used to handling in the real world.

Here, too, these restrictions force the interface designer to search for strictly visual ways to convey similarities and differences between objects, and to distinguish their various activities or states. Consider the endless discussions, in the first years of the Mac, of which icons should mean what, or the complexity (and ultimate unusability) of toolbars in complex applications like Microsoft Office.

An Object on the Screen is an Object in Storage (that is, it corresponds to a lump of stored data).

This metaphor is commonly applied when showing the contents of storage devices (or of their abstraction, file systems). This too builds on analogies to physical reality and is thus easily learned.

A subtle consequence of the application of this metaphor is that the user intuitively thinks that objects are unique, as they are in physical reality. In some contexts there may be several objects on the screen representing the same object elsewhere, as happens with "aliases" and "shortcuts". If these "reference objects" aren't distinguished clearly enough from "real" objects, the user may be confused as to what happens when he manipulates them; in any event, the lack of a real-life analogy usually demands an explicit explanation and rationale in the documentation.

A Folder is a Container, and Folders are Objects.

This is a complex metaphor, building on the previous two metaphors. Again, this is usually applied in showing storage devices, which are now recognized as special cases of folders — being containers of data which can actually be manipulated both on-screen and in the physical world.

An important point here is that this metaphor can be applied recursively. Unless the user has a penchant for routinely putting smaller envelopes inside of larger ones, or collects Russian "Babushka" dolls, this is a rare occurrence in real life, but quite common in the on-screen world. Again, some less imaginative beginners will require an explanation of this.

Note that the object-container relationship is not usually implemented literally on the storage device, but as a separately kept (and hidden) hierarchical data structure. Although this naturally leads to various display alternatives for the hierarchy, as indented outline-like lists and so forth, this discrepancy between displayed form and implementation often causes confusion when the designer attempts to lead the metaphor beyond its useful bounds. New users often have difficulty in understanding the “Network Neighborhood” display in Windows, since it conflates entities which are both physically and conceptually distinct.

Icons are Documents, and Icons are Programs

This is a superficially simple specialization of the “Icons are Objects” metaphor. However, at this point the metaphor is warped by the von Neumann paradigm which, at the same time that it proclaims that programs and documents (or data files) are both lumps of bits, stored in the same space, also maintains that they are very different: programs are “executable” or “intelligent”, whereas documents are not. This dichotomy already existed very early in the history of computing.

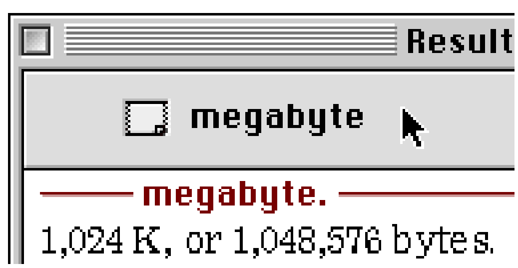
Not surprisingly it is here that many controversies arise. Some people argue that user interfaces should be “document-centric”, that is, programs as such should exist only behind the curtain, and the user should manipulate only documents. OpenDoc was an attempt to implement this paradigm. However, most current GUIs are either outright “program-centric” or try to hide the fact that they are.

Certainly naïve users can easily relate an on-screen “document” to previous experience with papers and books. However, there’s no easy real-life analogue to what a “program” is or does. To make things worse, modern GUIs try to confuse things by making something happen when both types of icon are “opened” – in one case, the program itself is run; in the other, the program associated to the document is run on the document. Explaining the last sentence to a novice user is not for the faint of heart, as nearly every word is used in a non-ordinary way.

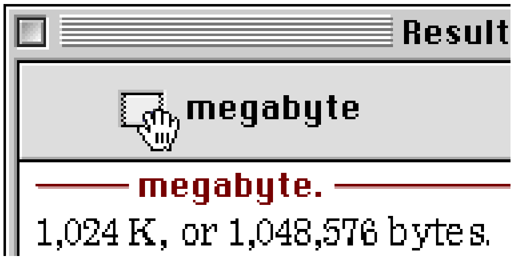
Needless to say, this offers endless opportunity for debate about the finer points of what a “document” is (or even if it really should be called by that name), whether the concept of a “program” is intuitive or not, and when one gets to windows, menus and so forth the whole debate tends to resemble medieval theology.

A Specific Example

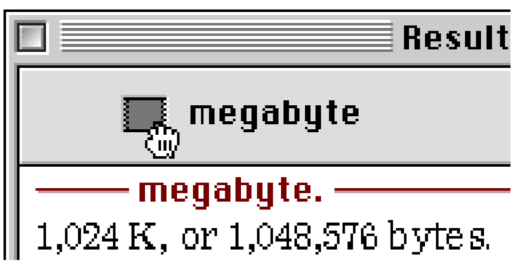
Let’s inject some life into the discussion with an actual application of some of these metaphors, taken from one of my recent programs — a dictionary application. Let’s suppose you have just looked up the word “megabyte”. A window with the definition appears, and over the text there’s a shaded portion (termed a “plaque” in Apple parlance) which looks like this:



Now let’s point at the small “clipping” icon with the cursor. This exercises the “cursor is a finger” metaphor in such a natural way that one ceases to notice it in a very short time. The cursor shape changes to a hand:

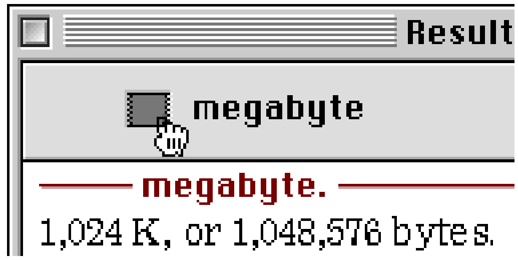


This shape change signals us that the cursor is currently over a draggable object, substituting for the real-world experience of actually touching an object (and perhaps lifting it slightly to see that it isn't attached to anything else). We're also, naturally, applying the "icon is an object" metaphor, which is facilitated by the distinct visual appearance of the icon. For experienced users, cultural familiarity with the clipping icon in other contexts also naturally leads them to this conclusion. Now we press and hold the mouse button:



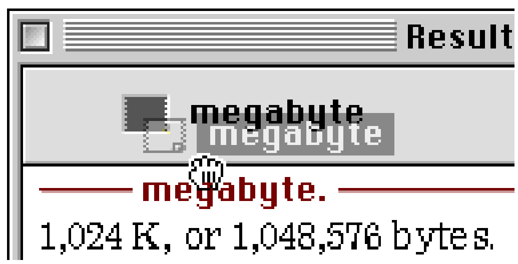
Notice that two further visual changes have occurred: the cursor shape has changed again, to a "closed hand" shape — indicating that something is being held in the metaphorical hand — and the clipping icon has darkened, to indicate that it is the object being held. Both these changes substitute for real-world kinesthetic feedback of grasping and holding an object.

So far the cursor hasn't been moved from the place where the "mouse-down" event occurred. Let's drag the cursor two pixels to the right and two pixels down:



Except for the cursor position, nothing has changed! Why? Well, the user may not have an absolutely firm hand, and / or may not have intended to actually drag the icon anywhere. Therefore, I built in a threshold below which no drag is considered to have started. At first glance, this may be counter-intuitive, but it's actually analogous to friction and / or inertia in the real world.

In this case, the threshold is 4 pixels, so now let's drag the cursor with more conviction to the lower right:



Several things happened here to give the user proper feedback to the effect that he has started a drag. You'll have to take my word for it, as you're reading this on a static medium, but be assured that:

- As soon as the movement exceeds four pixels in any direction, a soft "thuck" sounds.
- A translucent image of the clipping icon appears at the proper displacement from its original position, and a similar image of the search word appears at the proper distance from the icon, to emphasize that the definition of "megabyte" is being dragged.

- The clipping icon is replaced by an image of its empty, darkened contour, to show that something has been dragged from its starting place.

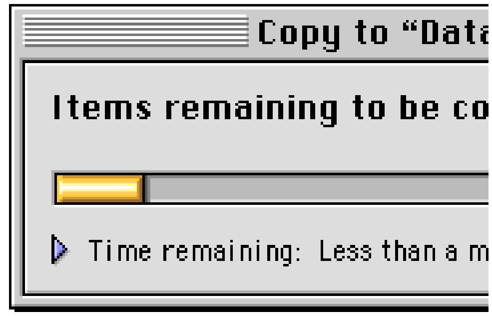
Now that the drag has been started, the you may drag the clipping anywhere you wish, and a suitable action will happen when the mouse is released; a clipping file will be created on the desktop, or the text will be inserted in another window, or the clipping will "snap back" to its original position should the destination be unsuitable for accepting the data. The actions above relate in a reasonable and easily understood way to the physical experience of moving an object from one place to the other — with the important but easily learned distinction that in this case, data are copied, and not moved.

The end result is that the icon resists movement at first, but then "pops" out of the plaque in a satisfying way. The whole effect is done in a life-like way without being obtrusive or especially noticeable at first glance. This was achieved by trying out various shadings, degrees of translucency, distance thresholds and sounds. One user commented, "I like dragging a clipping to the desktop, but I don't really know why!" There can be no greater satisfaction for an interface designer.

Not too coincidentally, this implementation applies another, less commonly used, grounding metaphor: "distance is tension". The increasing distance is "felt" by the user as increasing tension, which is finally released when the threshold is exceeded.

Amount of Substance is Amount of Time

This metaphor can be commonly seen in "progress bars":



The bar is progressively filled in as time passes, to indicate how far the task has progressed. Of course, the total time needed for the task must already have been computed, allowing a proportional part of the bar to be filled in.

There are examples of progress bars where the filled-in part is not proportional to time, but rather to data volume, or to the number of processed items. Especially in the latter case, there may be a large discrepancy between what the user considers reasonable and the actual movement of the bar. There are software installers which move the bar at varying speeds, giving at times the false impression that installation will finish quite soon, and at other times the impression that the system has crashed.

Other installers show two progress bars, the upper one moving in discrete steps according to the number of items, the lower showing actual progress for each item; when there are many short items, this will be extremely distracting, as the lower bar will just flash continuously. On Windows installers, there's usually the additional "misfeature" of progress bars formed of discrete rectangles, which directly contradicts our experience that time is a continuous process.

Even worse are progress bars that jump back to a previous state when the software concludes, belatedly, that it will take more time than originally projected. One should either tie the bar explicitly to data volume or some other predictable variable, or not use the progress bar metaphor at all. Having the bar jump back has no counterpart in physical reality, and the other alternative — having the progress bar itself lengthen proportionally to total time — is usually impractical in view of fixed window and screen sizes.

Of course, there will be instances when the total time cannot be estimated at all. In such cases, one should use the "spinning barber pole" or "chasing arrows", which simply denote that something of indefinite duration is happening. The "hourglass" and "turning clock" cursors are also variations of this visual indication.

Complex Metaphors are Left as an Exercise for the Reader

Once simple metaphors, such as the ones we examined above, are absorbed by the user, the complexity of modern computers demands simple ways to do complex tasks. In the real world, activities of the same level of complexity are either deconstructed into purposeful sequences of simpler actions (building a shelter) or are highly ritualized agglutinations of individually meaningless actions (courtship and other social activities).

Not surprisingly, user interfaces follow similar patterns. The user learns to do larger tasks in terms of already mastered smaller ones, or memorizes some sequence of actions which individually mean little but as a group have the intended consequences.

Rather than get into a detailed discussion of things like the use of color, buttons, windows and menus, I'd rather point out that, however satisfying or logical a particular implementation may seem to its designer (or to a satisfied user of that particular implementation), it may be viewed as essentially arbitrary; either *per se*, if the designer was careless, or in the derivation path from our grounding metaphors, if not.

A "good" user interface is therefore more akin to a work of art. Indeed, in the literature, terms like "elegance", "depth", and "satisfaction" are often found, which indicate that using a GUI is also an esthetic experience. In my opinion there is a strong correlation between the quality of this experience and the consistency with which our grounding metaphors are invoked.

Conclusions

Lakoff and Nuñez write (referring to mathematics): "…it arises from a wide range of common bodily experiences, makes use of the full range of the imaginative apparatus of the mind, has been actively constructed to serve important human purposes, and has come into being socially, over time, in the crucible of active debate."

All this applies as well to graphic user interfaces, with an important difference: they have been in the "crucible of active debate" for a comparatively short time. Hopefully, with new tools from cognitive science, we'll soon see more innovations in this important field.

Acknowledgements

Many thanks to Neal Stephenson for writing some of my favorite books; to John Lawler for reminding me of Neal Stephenson's essay, and for very kindly reading an early version of this paper; and to my friends Háj Ross and Stan Kelly-Bootle, for encouraging me to write for publication, and for catching many errors in my first draft.

[1] Lakoff, G., and M. Johnson 1999. *Philosophy in the flesh.* New York: Basic Books.

[2] Lakoff, G., and M. Johnson 1980. *Metaphors we live by*. Chicago and London: University of Chicago Press.

[3] Johnson, M., ed. 1981. *Philosophical perspectives on metaphor.* Minneapolis: University of Minnesota Press.

[4] Lakoff, G. 1993. *The contemporary theory of metaphor.* In A. Ortony (ed.), *Metaphor and thought*, 2nd ed., (pp. 202-51). Cambridge: Cambridge University Press.

[5] Lakoff, G. 1987. *Women, fire, and dangerous things: What categories reveal about the mind.* Chicago and London: University of Chicago Press.

[6] Lakoff, G., and R. E. Nuñez 1997. *The Metaphorical Structure of Mathematics: Sketching Out Cognitive Foundations For a Mind-Based Mathematics.* In Lyn English (Ed.), Mathematical Reasoning: Analogies,Metaphors, and Images. Hillsdale, NJ: Erlbaum

[7] Lawler, J., and E. Breck 1998. *Embodying Arithmetic: Counting on Your Hands and Feet.* Paper presented at LANGUAGING 1998. http://www-personal.umich.edu/~jlawler/embodying.doc.

[8] Lakoff, G. 1996. *Moral Politics : What Conservatives Know That Liberals Don't.* Chicago and London: University of Chicago Press.

[9] Lawler, J., *Metaphors We Compute By.* Original lecture delivered 1987, University of Michigan. In Dona J. Hickey (Ed.) 1999. *Figures of Thought: For College Writers.*. Mountain View, CA: Mayfield Publishing.

[10] Harris, R. A. 1993. *The Linguistics Wars.* Oxford University Press.

[11] Stephenson, N. 1999. *In the Beginning Was the Command Line.* http://www.cryptonomicon.com/beginning.html.

[12] Ramachandran, V. S. 1995, *Mirror Neurons and Imitation Learning as the Driving Force Behind "The Great Leap Forward" in Human Evolution* http://www.edge.org/3rd_culture/rama/rama_index.html.

[Editor's Note: Hofstader, Douglas R., *Analogies and Roles in Human and Machine Thinking*, Scientific American, September, 1981.]