Beyond metaphors and icons: towards a perception-action model for graphic user interfaces

May 30, 2007

Claudio Vandi (claudio.vandi@gmail.com)
Collegio Superiore Università di Bologna
Dip. Discipline Semiotiche

Abstract

How do we effectively understand graphical interfaces in order to act in the virtual ambient they create? Metaphors in user interfaces are meant to give the user a familiar context in which to act: an office with recycle bins, desktops, folders and files. The problem is that incoherence lies everywhere if we look beneath the surface.

In this paper I will try to show that metaphors are used to create a semiotic ground on which single actions can emerge. Where a more classical analysis of metaphors in graphic interfaces would focus on the single metaphor or on the system created by the semantic fields, I focus on a more holistic figure/ground dynamics. Smaller-than-the-single-metaphor elements constitute the ground that allows the user to make action oriented figures emerge. The tools I develop to describe this are grounded in the theory of semantic perception (Rastier) and in the semiotics theory of C.S. Peirce.

My objective is to find a model of the user as someone continuously interpreting and acting in the interface. The interface should support its actions and not force him to create complex hypothesis about icons and visual elements and their possible employment. To show how design influences the user actions I will use examples from Mac OSX and from Windows XP and its recent Vista development.

1 Introduction

To analyze objects of design in semiotics usually means two things:

brand analysis: analyze how a product can be set off by its difference from similar objects;

content analysis: analyze what are the meanings an author meant to put in some object he designed; (Floch, 1995).

What I will try to do in this paper, on the contrary, is to show how semiotics can go beyond the “look and feel” to analyze the user’s action in its practical unfolding. I will do this by rethinking the usefulness of metaphors in Graphic User Interfaces. My point is that if we choose our categories wisely, then semiotics can say something on how design matters in the continuous action of a user carrying out some task. I will start presenting one of the most interesting theories on semiotics and Human Computer Interaction.

2 The semiotic engineering

In *The semiotic engineering of Human Computer Interaction* Clarisse De Souza shows how semiotics can enter the design process developing a theory of interaction as a communication process:

> The semiotic engineering of HCI artifacts is about the principles, the materials, the processes, the effects, and the possibilities for producing meaningful interactive computer system discourse (DeSouza, 2005).

Behind the monitor we have a writer (the designer) with an idea of the final users of its product, which includes the possible tasks the user can perform within the software and his semiotic competences. In front of the monitor we have a reader (the user) that has to correctly interpret what the designer wanted him to understand by putting that sign in that place. For example he will understand the “cut” operation by recognizing the scissors icon and its pertinence in that context (cutting the text). I quite agree with this project. In a certain way we have to understand and recognize something in order to use an interface. But, if we take the user’s point of view we also have to sketch a theory of how the user learns the interface while using it, and not only by recognizing the referent of a single sign. This is the difference between a semiotics of signs as taxonomy and a semiotic of signs as action oriented elements. To be used, to be “acted”, is the key role of interfaces. Pragmatics teaches us that there is no meaning out of a pragmatic horizon, out of an actual practice. This leads to a semiotic theory that can say more than what is needed to recognize a sign as a sign of something. The semiotics I’m thinking of is the one created by **C.S. Peirce**, from which I will take some central concepts: *semiosis* as a continuous pervasive process already active in perception; *habits* as “a general law of action, such that on a certain general kind of occasion a man will be more or less apt to act in a certain general way” (C.P. 2.148) and *diagrams* as non static sets of relations that “represents the probable course of experience” (C.P. 2.169) and that, unlike mentale images, can be manipulated for action (Peirce, 1966).

I will discuss these ideas by first doing a wider critics to metaphors.

3 Metaphors

Metaphors are one of the central elements of HCI design since the first Apple Guidelines (Apple-Computer, 1984). Their widespread use 20 years after should account for their usefulness. Metaphors in user interfaces are meant to give the user a familiar context in which to act: an office with recycle bins, desktops, folders and files. The core idea is that we can transpose competences of the real world in the virtual ambient thanks to a mapping process (Rohrer, 1997). We know how to use a trash in our office, thus we know how to use a virtual trash on our desktop. But are those metaphors so intuitive? Why my wastebasket is on the desktop, not under it? Why can we trash files, music, images and restore them without loosing the original quality? That is not at all what happens with real wastebaskets and documents. Furthermore: why we open a window by clicking on a folder located on the desktop? And why that window has folder in it? Windows shouldn’t be transparent? In the graphic interface they are not, but they gain a new property: they are magically resizable.

We can conclude that the metaphors used in the "desktop" interface presents an intuitive environment to the experienced user, but an alien environment to the beginner. I know that the critic of metaphors is not something new in the HCI debate. Almost 10 years ago, in one of its basic works on design, Donald Norman provided some good arguments for designers to forget the word "metaphor" (Norman, 1998).

What I'll try to do here is to show how we can get rid of metaphors without loosing coherence and its benefits.

4 How to save metaphors

What could be the benefits of these weak metaphors? My idea, that I will clarify by some examples, is that metaphors are used to create a **semiotic ground** on which single figures can emerge (Rastier, 1987). This ground proceeds from micro-elements (likeness, visual rhymes) to macro-elements (virtual objects) and is active on two levels. *First*, it constitutes the visible ground from which salient perceptual figures come out. *Second*, it supplies a conceptual coherent ground from which the user crafts task oriented figures. In this perspective the semantic ground is not something that we need to rely on while using an interface (we don't activate the "office" frame when we light on the computer) but is something that is created through a presumption of isotopy (Rastier, 1987) based on the regularity of superficial elements. The ground is at the same time the homogeneous ambient that makes possible to act similarly in similar situations, and the result of these actions:

> To perceive is always to sketch a sense that is not developed if not in a path, in an activity of thematization in which identities are made and unmade Visetti and Cadiot (2001).

This activity consists in recognizing elements belonging to the same domain and to act accordingly.

Going back to Peirce Isotopies allow us to create habits as path of repeatable actions. There are hundreds of possible actions that can be performed within a graphic interface (drag, drop, cut, paste etc.) and become strong competences for trained users. Metaphors provide a simple ground for superficial actions. The meaning of a metaphor doesn't depend on the recognition of the related real action (if scissors then cut). The meaning of a metaphor-icon is the answer we get while acting on it. As Norman says

> Something that happens right after an action appears to be caused by that action (Norman, 1988).

Coherence (the structural force of metaphors according to Lakoff and Johnson, 1980) is then built when we get a similar answer while acting in similar spaces or objects. To cut a string of text will create the same answer than to cut a music file or an image. This answer becomes the extensive metaphoric meaning of "cut" even if the real action performed by the computer is very different each time. The "metaphor name" is a false problem: we could call the window "drawer" if we think that this better portraits it, but what the drawer does on the desktop should be done also in an application window.

5 Some examples

Let's go back to design: the User Centered Design (UCD) dogma is "identify as precisely as possible what the users want and need" (DeSouza, 2005). The computer should always work without asking the user to make a big cognitive effort. A good designer should have a good idea of what the users want to do by using a program, and should craft the interface consequently. This is a good point: one of the basic lessons of Eco's semiotics is that you have to build a model of your reader if you want some text to be incisive (Eco, 1979).

But we could raise some doubts about whether the users themselves always know exactly what they want to do, and if it is possible to consider users as an homogeneous group of people. The diffusion of computers should account for the difficulty of the task. Try for example to think at Negroponte's 100 dollars computer: will we design cd players icons to signify "music"? Of course we won't. Furthermore, can we claim that people from Los Angeles to Taiwan are the same user? And can we say that when using a computer users already know what they want to do and only have to learn how to do it? If we take someone that has to write ten times the same sentence in a document and he does that by taping ten times the same key sequence, can we say he is correctly using a text editor? So my point is that the goal of a good design is to make the user's learning curve very steep. And do this with two steps:

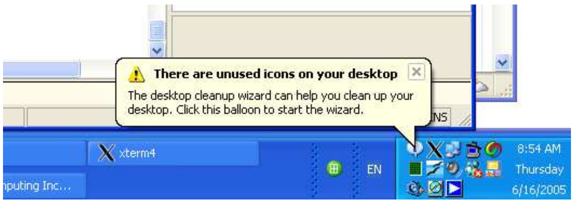
1. by giving him the competences on how to use the interface while he is using it;

2. by using the maximum of his previous competences to acquire the new ones;

My idea of "simple" is not simply that users "shouldn't be aware of using a computer" as Norman says (Norman, 1998), but that they shouldn't be aware of the fact they're using previous complex knowledge to carry out their tasks. There are two ways to do so. I will identify the first as the Window's strategy, and the second as the Mac strategy. Of course both the Operative Systems have similar elements, but my point is that they have opposite development directions.

5.1 Windows

The Windows's strategy consists in asking the user what he wants to do at every step, offering the solution the designer thinks the user is looking for. Something like saying "I have this model of you that makes me argue that if you are doing this you want to do that, am I right?" This is something closer to divination than to design. When the system relentlessly asks "do you want me to remove unused icons from your desktop" (img. 1) it is not just offering something but it is asking us to make it do something. For the modalities lovers the "I want that you want to make me do something" recalls the masochistic willing more than the designer's attitude.



There are unused icons on your desktop

The desktop cleanup wizard can help you clean up your desktop. Click this balloon to start the wizard.

img. 1

In any case, our answer is more often "don't bother me" than "yes thanks". And as the system rises periodically the same question we do always the same movement:

- *Stimulus* Yellow balloon

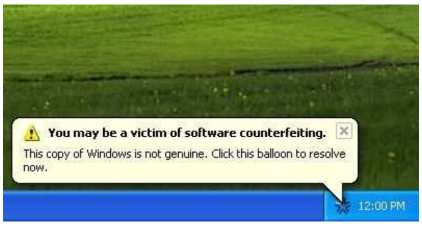
- *Response* Move toward the down right corner and click **x**

Users really hate this kind of assistance:

> The reason I put icons on my desktop is BECAUSE I don't use the program very often. If I use it, it'll be in the start menu, or I'll put it on the quick-launch bar
>
> This is one of the most scary 'features' of Windows. Can we turn it off at all? It keeps pestering me... (*http://www.weiqigao.com/)*

What we have to do in Windors is to learn a **negative habit**, an habit we get without being aware of, in order to get rid of something. This is insane. And it is even more so if we think that messages deeply differ in their style: we pass from "would you like me to help you cutting your text?" to something like "error 1234 in key xyz". So the user has to understand every time the message style (whether it concerns an error, or a suggestion) and then he can act consequently. This means not to know how to take advantage of the user's habits.

It is worthy to note that Windows acts in the same way when he wants to prevents us from using it:



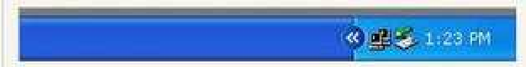
You may be a victim of software counterfeiting.

This copy of Windows is not genuine. Click this balloon to resolve now.

img. 2

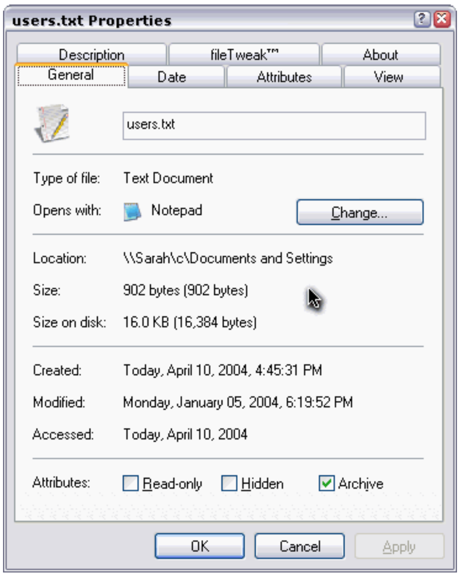
More designer's mistakes (thinking of being helpful)

Taskbar shows only frequently used elements. When you use something new it enters the taskbar and shifts the other icons behind the < < sign.



img. 3

In many "properties" windows, tabs are on two or three lines. When you click on something on the third line it jumps on the first and the overall configuration changes. What was on the third is now on the first. The problem is that "being the third at the top right" is not an accident, it is a constitutive part of the element. As well as colours and names, it gives the user the possibility to move toward something he knows as being there. Changing the arrangement every time doesn't help: it forces to look and understand every time before acting.



users.txt Properties

users.txt

Type of file: Text Document

Opens with: Notepad

Location: \\Sarah\c\Documents and Settings

Size: 902 bytes (902 bytes)

Size on disk: 16.0 KB (16,384 bytes)

Created: Today, April 10, 2004, 4:45:31 PM

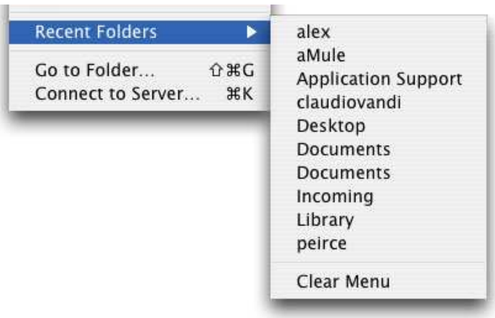
Modified: Monday, January 05, 2004, 6:19:52 PM

Accessed: Today, April 10, 2004

Attributes: □ Read-only □ Hidden ☑ Archive

img. 4

To be fair (and because Apple doesn't pay me) here is a similar bad example from OSX. In the finder menu you have a "recent folders" submenu that lists the last ten folders you have visited. I find this very useful while working through multiple sources. OSX lists the folders alphabetically. This is also good because it is a stable criterion, no matter what you have visited last.



img. 5

But this good idea doesn't work in the case you have two folders with the same name: let's say Work -> *Documents* or Home -> *Documents*. First of all you are not told which which folder the subfolder belongs. So you have to try clicking on one of the two. Let's say you click on the second. When you come back to do that again the second one would not be the same as before, because by clicking it you have made it the most recent folder. The criterion to list similar elements is "the last you visited is listed first". This is bad design. The system holds memory of the user's action but in a wrong way, instead of

> have an evolutive representations of its user, becoming richer every time he acts on it (Rastier, 1991).

it outsources cognitive skills: it is in fact the user's duty to remember what was the last folder he visited. A simple habit (first one, second one) is turned into an inference (the first one is the last I've visited: which one I've visited last?).

I have three points for saying that these are bad ways of helping the user:

1) they compel the user to have negatives habits;

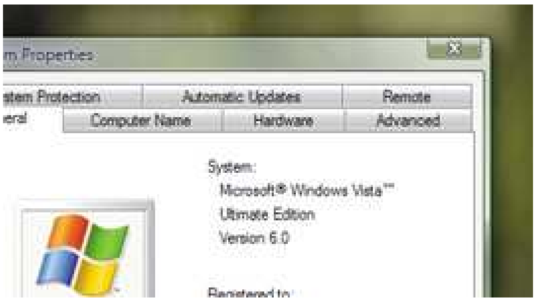
2) they stop the user's action by using meta-communication messages. Every time a message from the author gains the scene the user has to stop and come to terms with the fact that he is using a computer;

3) they encage the users in compulsory narrative programs, where single stages are not crafted by them but created by the computer. This is about bad narration: if the narrative program is compulsory there will be no novelty in reading the story;

5.2 "Vista's revolution"?

Now, if you take a look at the new Windows Vista it is quite clear why a *look and feel* restyling is far from a design rethink.

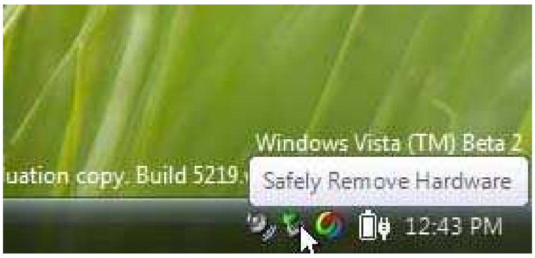
Properties windows are the same



System:
Microsoft® Windows Vista™
Ultimate Edition
Version 6.0
Registered to:

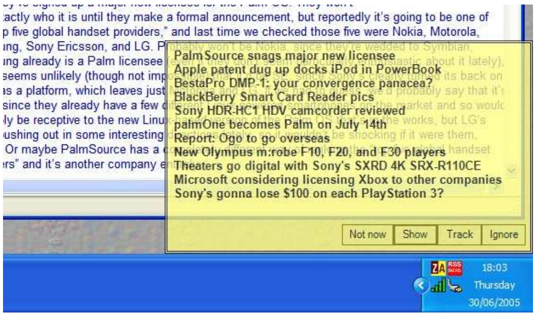
img. 6

Pop-ups proliferate


Windows Vista (TM) Beta 2
uation copy. Build 5219.
Safely Remove Hardware

img. 7

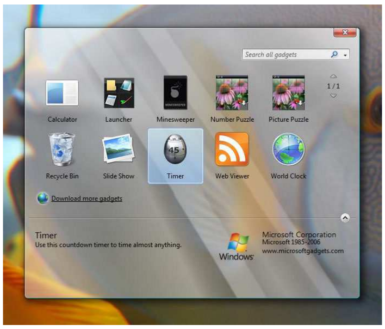
until they almost replace alert messages


PalmSource snags major new licensee
Apple patent dug up docks iPod in PowerBook
BestaPro DMP-1: your convergence panacea?
BlackBerry Smart Card Reader pics
Sony HDR-HC1 HDV camcorder reviewed
palmOne becomes Palm on July 14th
Report: Ogo to go overseas
New Olympus m:robe F10, F20, and F30 players
Theaters go digital with Sony's SXRD 4K SRX-R110CE
Microsoft considering licensing Xbox to other companies
Sony's gonna lose $100 on each PlayStation 3?

img. 8

This could be a good "damage reduction policy", because you only have to learn a negative habit: bothering messages are in that position and have that look. But they don't disappear, you have to make them disappear. Now they are transparent. Semiotician would probably say they are less obtrusive, but users don't think so.

New windows are transparent elements, they are not the desktop but not separate from it, they have a liquid look, but some sort of a shadow.



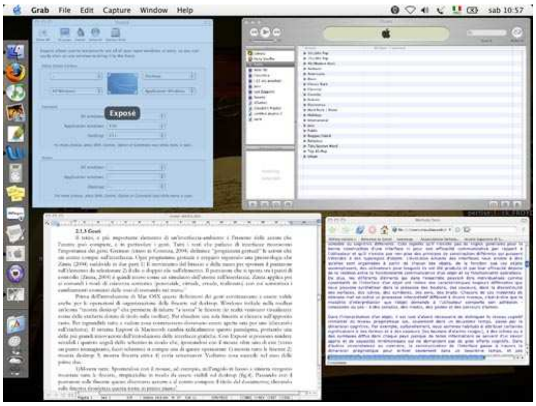
Timer
Use this countdown timer to time almost anything.

img. 9

The Vista main feature is really a “Wow” one (do you remember the commercial?). Can you really say you see something more than the first window? Is it someway useful? You say “Wow: but where is my window?”.

img. 10

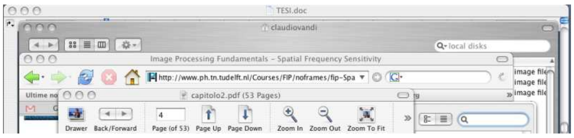
Compare it with *Exposé* in Mac OSX



img. 11

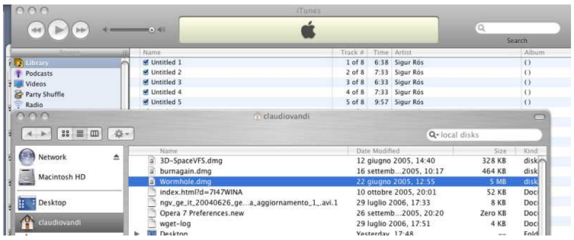
5.3 Mac OSX

In Mac OsX we have three very different objects called “window”: the application window, the folder window and the browser window. To strengthen the connection offered by this common name they share different graphic elements: the shadow, the keys, the buttons and the colors. It is of course nonsense calling them by the same name and this has proved to confuse the user under certain circumstances. On the other hand the presence of these elements creates a common ground to act in a similar way in the different areas and situations. Shadows and illumination (see img. 5 above) makes evident the position of the window on the visual ground, in Mac OSX you don’t have full screen windows like in Win, every window is a single element separate from the desktop. OSX don’t replace the desktop creating a new global and different ambient as in Windows, but introduces figures where buttons and windows shape allow the user to rely on previous habits. Windows are individual but like figures.



img. 12

Indeed, the new developers philosophy promoted by Apple, which Steven Johnson calls “the swiss-army knife approach” (Johnson, 2002) goes in this direction. Even if the applications are tools for a specific domain of use (iTunes, iPhoto etc.), they share the location of common tasks buttons (close, magnify, delete) and the internal display of elements (icons or lists of file names ordered by size, date etc.)



Name	Track #	Time	Artist	Album
☑ Untitled 1	1 of 8	6:38	Sigur Rós	( )
☑ Untitled 2	2 of 8	7:33	Sigur Rós	( )
☑ Untitled 3	3 of 8	6:33	Sigur Rós	( )
☑ Untitled 4	4 of 8	7:33	Sigur Rós	( )
☑ Untitled 5	5 of 8	9:57	Sigur Rós	( )

Name	Date Modified	Size	Kind
3D-SpaceVFS.dmg	12 giugno 2005, 14:40	328 KB	disk
burnagain.dmg	16 settemb...2005, 10:17	464 KB	disk
Wormhole.dmg	22 giugno 2005, 12:55	5 MB	disk
index.html?d=7I47WINA	10 ottobre 2005, 20:01	52 KB	Doc
ngv_ge_it_20040626_ge...a_aggiornamento_1_.avi.1	29 luglio 2006, 17:33	8 KB	Doc
Opera 7 Preferences.new	26 settemb...2005, 20:20	Zero KB	Doc
wget-log	29 luglio 2006, 17:51	4 KB	Doc
Desktop	Yesterday, 17:48	--	Fold

img. 13

It is not just aesthetics, it is **pragmatics**. The user who doesn’t understand how to delete a file from a new application he is using, will be helped by a “recycle bin” icon. He will do that because the same perceptual elements led him to a success in a previous situation, not because he recognizes the real referent of the metaphor. The perceptual similarity will prompt him to do it, by smoothening his potential feeling that this is a new and different situation. Far from being pure semiotics slang this idea is the formalization of a principle that is well known among Interaction Design experts:

> Most of the elements of intuitive graphical interfaces are actually visual *idioms*. We understand the idiom simply because we have learned it and because it is distinctive, not because we understand it or because it makes subliminal connections in our minds (Cooper A. Reimann, 2003).

Apart from the word i"diom", which in linguistic refers to something shared among a community, this quote gives a good idea of how the importance of habits is known among interface designers.

Summarizing linking these examples with the theory sketched above: coherence between elements is replaced by isotopies, based on finding similar elements, and on the possibility of doing similar actions in similar spaces. Semantic coherence is thus replaced by a perceptual coherence and by a pragmatic, action-oriented, one. The task of a good design is to use these elements to stimulate the learning by the easy creation of habits of action.

6 Open conclusion: affordances

I would like to conclude with a passage through affordances which have a key role in User Centered Design, and allow me to reconnect single parts of my paper under a design perspective. The first who had the idea of applying affordances to design was Norman in 1998

> affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used (Norman, 1988).

And later explains the power of this idea applied to design:

> The real question is about the perceived affordance: Does the user perceive that clicking on that location is a meaningful, useful action to perform?(Norman, 2004) .

Norman's idea is that when simple things need explanation design has failed. But two doubts can be raised about the use Norman makes of affordances. First one could claim that, in reducing affordances to the mere visual qualities of objects, Norman simplifies Gibson's concept. I agree with this critic, but I think that we should look for affordances where they were first born: in the Gestalt theory:

> The dimension of *requiredness* in Kohler, of *Aufforderungscharakter* in Lewin, or *affordances* in Gibson [...]: there is a constitutive solidarity between objects and practical routines; objects and practical fields are even mingled with action projects in which they are kept (Visetti and Cadiot, 2006).

This is my project for the future, and I won't go deep in the subject now. Secondly, Norman is also aware of the fact that every object has many affordances, some of them yet to be discovered. So, it would be a good idea to enquire how we select one action or another and how affordances become stabilized in some practice. Again, Peirce can help us. Reconsider what we have just said: when you see something you don't see a neutral thing (there is no *degrée zero* unhappily for Barthes) but you see an object already imbued of relations to what you are doing (your practice) what you already know of that object (pragmatics and habits) and the action you foresee to perform on it (diagram). You are ready to use it because of diagrams of possible actions and because of habits that select the right one, not because it tells you something about itself. As for affordances some knowledge of an object can be transposed to another and (not so) intuitively make us understand how to use it in a similar practice.

If affordances make us perceive that some action is possible, how do we choose that action among the others? In Peirce's words and to conclude:

> We imagine cases, place mental **diagrams** before our mind's eye, and multiply these cases, until a habit is formed of expecting that always to turn

out the case, which has been seen to be the result in all the diagrams. To appeal to such a habit is a very different thing from appealing to any immediate instinct of rationality. That the process of forming a habit of reasoning by the use of diagrams is often performed there is no room for doubt. (CP 2.170 The Fallibility of Reasoning and the feeling of Rationality).

References

Apple-Computer (1984). *Macintosh Human Interface Guidelines,.*

Cooper A. Reimann, R. (2003). *About Face 2.0: The Essentials of Interaction Design.* Wiley.

DeSouza, C. (2005). *The Semiotic Engineering of Human-Computer Interaction.*

Eco, U. (1979). *Lector in fabula.*

Floch, J. (1995). *Identites visuelles.* PUF, Paris.

Johnson, S. (2002). Is the computer desktop an antique? *http://www.slate.com/?id=2075219.*

Lakoff, G. and Johnson, M. (1980). *Metaphors we live by.* Chicago University Press.

Norman, D. (1988). *The psichology of everyday things.* Basic Books, New York.

Norman, D. (2004). Perceived affordances. *http://www.jnd.org.*

Norman, D. A. (1998). *The invisible computer.* The MIT Press, Cambridge (MA).

Peirce, C. (1958-1966). *Collected Papers.* Cambridge: Belknap Press of Harvard University.

Rastier, F. (1987). *Semantique interpretative.* PUF, Paris.

Rastier, F. (1991). *Semantique et recherches cognitives.* PUF, Paris.

Rohrer, T. (1997). Conceptual blending on the information highway. *http://zakros.ucsd.edu/ trohrer/metaphor/iclacnf4.htm.*

Visetti, Y. and Cadiot, P. (2001). *Pour une theorie des formes semantiques.* PUF, Paris.

Visetti, Y. and Cadiot, P. (2006). *Motifs et proverbes.* PUF, Paris.