

Interfacing by material metaphors

How your mailbox may fool you

Marianne van den Boomen

Operating a computer is quite easy nowadays. We all click our way through our desktops; we send mail; we save documents in folders. We barely realize how these actions are framed by metaphors. Our first association with ‘cleaning a desktop’ has probably more to do with deleting unused icons and files on our PC than with polishing our material desk. We have no problem with the dominant office metaphors, even when they blend with metaphors from other settings, such as home, menu, or window.

Operating these metaphors usually goes smoothly. But, as we all know, sometimes things go wrong. Sometimes our actions do not yield the expected results, and our computer suddenly becomes a black box. In this chapter I aim to trace what happens when we operate our PC, and how metaphors enable or disable our access to the machine. I will take a look at the user interface and show how it may fool us with its icons, begging the question: what exactly are icons? Are they signs, are they tools? With a little help from Peirce, Heidegger and Hayles, I will end with a plea for the development of a theory of material metaphor.

A computer is no coffee machine

An ordinary PC is in fact an extraordinary apparatus. It is not only a physical machine, able to perform labor, i.e. to transform energy and raw material into a product or an event; it is also a symbol-processing machine (Hayles 2002; 2005). As a symbol-processing device, it translates its internal physical states into machine code as well as human code – code readable by humans. The latter is usually represented on the screen by the graphical user interface (GUI) with its icons, menus, and windows. Here, the PC articulates itself as a set of tools for the production, reproduction and modification of data products (texts, images, soundtracks). At the same time the PC is also a toolmaking machine, able to create new tools (programs, scripts, plug ins) which can be fed back into the very same machine. This makes the PC a much more complex device than just a set of tools.

At first sight, tools and products are clearly differentiated on the user interface. Products exist as mutable data objects (files); and tools exist as executable sets of commands (programs), or as interfacial signs (icons, buttons, menus). However, at a deeper level these tools and products are intricately nested into each other. Executable programs also consist of sets of files; interfacial signs are representations of executable commands, and thus they are, in the last instance, also files. This complex machine of nested files becomes even more complex when hooked up in a network with other computers, thereby also enabling transference of tools and products to other machines in the network.

Yet, users stick cognitively to the assumed clear division between tools and products. After all, means and ends, tools and products, are ontologically quite different, as is the case with a coffee machine: there is a clear difference between the button for cappuccino and the cappuccino itself. We never mistake the button for the coffee. But a computer is no coffee machine. And it is actually quite common in computer praxis to mistake the button for the product itself. This is due to the fact that we cannot have the product without the tool. We cannot read a text document without a word-processor program; we have no access to our mail without a mail program.

In order to work with this compound, and differentiate whenever necessary, the usage of the keyboard and the mouse is decisive; these hands-on devices are indispensable parts of the user interface. However, the visual screen with its metaphorized tools and products seems to attract all the attention. The screen seems to integrate, supersede and cannibalize all other parts of the user interface. This engulfing mechanism is not limited to computer usage as such; the screen pops up as a dominant metaphor for digital media in general, in ordinary speech as well as in new media studies (Turkle 1995; Johnson 1997; Bolter and Grusin 2000).¹ This may be explained by the ability of the GUI to translate and articulate all other components into visual representations, thus rendering irrelevant what remains invisible, and rendering unthinkable what remains unmetaphorized.

The visual tools on our screen thus seem to work as metaphorical stand-ins for complex machine processes. After the metaphorical translation of machine code into human code, there seems to be no way back. Metaphorical sign-tools acquire a life of their own, cannibalizing everything else that might be there. By showing they hide, by translating they substitute.

This mechanism is especially at work in computer icons. These small pictures are shortcuts to specific software commands in order to yield some result through the combination of human and machine actions. However, machine actions are rendered invisible wherever possible. Icons cloak their reference to software instructions and machine acts; instead, they pretend to refer to particular 'places' on a computer (My Documents, My Network, mailbox, folder) or to particular 'things' (file, program). Sometimes it is not even clear whether an icon refers to a place, a file, or a program. Take for instance your mail icon. What does it stand

for? 'Well, my e-mail of course.' But what exactly is implied in the conceptual shortcut 'my e-mail'; does it refer to a specific program running? Or to a place on your computer, the mailbox, where your mail resides? Or to a set of files, sent to you as e-mail messages? Or perhaps to all of these in one?

Usually, when your mail just works as expected, these questions are irrelevant – who cares? But sometimes these questions pop up, as in the following story. It shows how different references may converge or diverge, leading to a lot of user bewilderment.

Somewhere around 1997 a friend of mine had problems with her brand new broad-band connection. She asked me to have a look at her computer. When she clicked on the mail icon an empty inbox showed up.

'See, no mail. And I'm sure I have mail; I forwarded some from my office. That man who did the installation yesterday said everything was working – just click and go, ma'am. But where is my mail?'

'Okay,' I said, 'First, are you sure you are online?'

She pointed vaguely at the wires, still hanging chaotically all over the place, swirling to her desk. I hoped they were okay, but I did not mean the cables. There was a connection icon on her desktop, displaying a telephone device.² Fortunately, its settings turned out to be correctly configured; a double-click established the connection with her Internet provider. And after clicking the 'get mail' button in her mail program her mail streamed into her inbox.

'That's it,' I said. 'You first have to connect, with that icon.'

My friend was puzzled. 'That telephone icon? I used that to dial up with my modem, but now I have a permanent connection, just like I have at work. There I always see my mail immediately, without having to click on anything. I don't even have such a telephone icon there.'

I explained that apparently her computer at work was configured that way. 'That telephone icon stands for any kind of connection, be it by phone line or cable. But you can bypass it. It is possible to connect automatically to your provider as soon as you start up your computer. And also to fetch your mail as soon as you start your mail program. We can arrange that, if you like.'

And so we did.

That was all. A clear, comprehensible problem; just a small conceptual error. Yet, the story is illustrative of how human-computer interaction works, how it often fails to work, and how this relates to conceptual reifications, material configurations, and hidden steps in between.

Basically, my friend took the metaphorized icons literally. She took the mailbox icon for the mail itself, and the telephone icon for using the phone itself. For her, the mailbox icon was not referring to a process, a string of commands set in action in order to obtain a computed result; rather, it functioned as a key to a

specific place, her inbox, where she expected her mail to be, immediately. The icon was not read as a reference, but as the immediate referent itself. The icon had superseded and cannibalized all references, transferences and network labor involved. A shortcut indeed.

It is important to note this is not unique to my friend's computer use. It is a general feature of iconicity. Computer icons function by iconic condensation: a condensation of reference, referent, and meaning into one visual metaphorical sign, thereby substituting all involved complexities with a stable icon. Hence, the inclination to take the icon for a specific *state* (result, place, or thing) instead of a referential button able to invoke a performative *process* is very strong. In fact, this is part and parcel of the very function of icons: shortcutting and condensing the signifying and executing processes. Icons can only carry out their signifying and executing job by concealing the involved complexities, substituting them with stable entities.

Hence, the mail icon hides the nested processes it refers to: executing the mail program, including its configurations for a particular Internet connection, a particular mail account, and particular incoming and outgoing mail servers located at an Internet service provider. Instead, the icon represents a specific result (received mail) located at a specific place (the mailbox), as an ontologized entity. The same holds for the telephone icon: it conceals that it is referring to the execution of a connection program, including its configurations and settings. Instead, it represents the contiguous telephone device.

This concealment of software and hardware processes cannot be seen as coincidental 'non-representing'; it is a necessary and deliberate act against representation. I propose to call this act of deliberate concealing *depresentation*. We could then say that computer icons do their work by representing an ontologized stable state, while depresenting the procedural complexity. This is the task we have delegated to icons. And this is why we are seduced, indeed compelled, to take icons literally – that is, to ontologize them, to condense and collapse them into their own inextricable iconic ontology, their *icontology*.

Signs, iconicity and indexicality

This begs the question: what kind of signs are computer icons? Peirce's semiotics (Peirce 1873) seems typically apt to solve the riddle of the icon, not only because it unfolds an explicit vocabulary on *iconicity*, but also because Peirce's notion of *indexicality* enables a non-essentialist analysis of the relation between signs and the world outside language and signs.³

Peirce's concept of indexicality does not pertain to just any reference a sign can make to an external object. Peirce distinguishes three basic relations a sign can have to the object it refers to: iconic, indexical and symbolic. The *icon* refers to its object by *resemblance, similarity or analogy*, for example the portrait of a person, a

traffic sign with a picture of a hollow in the road, or the map of a country. The *index* refers to its object by an *existential, physical or causal* connection, for example smoke indicating fire, a fingerprint identifying a person, a sundial indicating time. The *symbol* refers to its object arbitrarily, that is, by *habit, convention or law*, for example the flag of a country, the logo of a company, or the alphabet for phonemes.

These categories are not mutually exclusive; in a symbol there may be something indexical and something iconic. For example, a traffic sign is always a symbol, as it is taken up in a system of laws and conventions regarding the use of forms and colors for these signs. However, a particular traffic sign may be indexically referring to a nearby hollow in the road, and it may do so by an iconic image resembling a hollow. Peirce's classification also accounts for the fact that signs circulate and may change by use. Icons may become symbols, for instance when the iconic picture of Che Guevara is printed on a T-shirt, signifying not so much the person Guevara but radical left sympathies. Likewise, symbols may be reiterated as an index; a yellow letter M is a symbol referring to the brandname McDonald's, but as a sign on a pole it is indexical for the nearby hamburger outlet.

Peirce's dynamic triad may be able to account for the interaction between state, process and system at work in computing semiosis, especially when it comes to computer icons. At first sight, computer icons seem not to be Peircian icons at all, but rather Peircian symbols. After all, they are all arbitrary; what a specific icon stands for we have to learn by collateral experience. But at second sight, computer icons rather seem to be Peircian indices: they all refer causally and physically to a set of software instructions to be executed.

Meanwhile, on yet another level icons may articulate other relations since they can refer to their object in three ways: by resemblance (iconic), by existential relation (indexical), or by convention (symbolic). On this level the mailbox icon would be a genuine Peircian icon, since its relation to its displayed object is based on an analogy between postal mail and e-mail. The telephone icon would be an indexical sign: its representation is not based on analogy but on an existential (though outdated) relation with the telephone device. And the icon for Microsoft's Internet Explorer, the stylized 'e', would then be a Peircian symbol, since the image is arbitrary and conventional.

From one perspective, then, all computer icons can be seen as symbolical or indexical signs, and from another perspective they may be icons, indices or symbols. The latter concerns *how the object is represented* by the sign, the former pertains to the *external existence of the object*. In Peirce's terminology, these are different kinds of objects. He called the first one the *immediate object* – the object as represented by the sign – and the second one the *dynamical object* – the object as it exists in the world. This dynamical object cannot be expressed by the sign itself, it can only be indicated, and its connection to the sign has to be learned (Peirce 1909).

This differentiation between objects enables us to flesh out how computer icons work. As we have seen, desktop icons refer to an act of executing machine code. From this perspective, all desktop icons are indexical: they refer to existential, physical chains of causation, to machine processes to be executed. Their dynamical object is thus machinic code. Whatever their immediate object as expressed in the sign may be (an iconic mailbox, an indexical telephone device, a symbolical logo), their dynamical object is code.

But as we have seen, computer icons not only refer to machinic code but also to human code – readable symbols and metaphors. Hence, a computer icon indexically refers to machine code, and symbolically it refers to human code (e.g. ‘mail’, ‘connection’). In that regard, computer icons can be considered Peircian indices (referring to the dynamical object of machine code), wrapped in Peircian symbols (referring to the dynamical object of human code), while they are only contingently iconical (if they use pictorial resemblance for representation).

Yet, in the praxis of signification this gets inverted: computer icons establish themselves as primarily iconical. While they are rarely genuine Peircian icons, they all exhibit what I have called ‘iconology’ – reified iconicity. They do so by equating and substituting the sign with its immediate object of reference, thus negating the indexical and symbolical references to the dynamical object.

In order to understand the strange double dynamical object of computer icons – simultaneously machine code and human code – I propose a third kind of object: a symbolic future object, a not yet actualized object, an object to become, in short: a *virtual object*. This object is not represented in the sign, neither is it the indexical, actual execution of machine code. It is what *might be actualized* by the joint venture of human acts and machine processes. It is not just your mail program running, it is that you might have received mail. It is not just your browser running, it is the web out there where you may find or do something. It is where machine possibility and human effort come together. It is where signs become tools for unknown and unprogrammed virtual objects.

Tools ready-to-hand and present-at-hand

This introduces another problem: how can signs become tools? At first sight tools are fundamentally different from signs. Signs are marks able to qualify, refer to, or represent other signs and things in the world; tools are things in the world able to shape and transform other things. But as Heidegger (1927) pointed out, tools exist *Um-zu* (in order to); their being-in-the-world always refers to a possible (may we say ‘virtual?’) work to be done, taken up in a system of other equipment (*Zeug*) and labor. Heidegger calls a tool we just use *ready-to-hand* (‘Zuhanden’); it fits with the acts of our hands, and the goal in our minds. It aligns seamlessly with its *Um-zu* and does not call for further reflection – we do not think about a hammer, we just use it. But when a tool does not function according to its *Um-zu*, for

example when it breaks down, it suddenly becomes a strange separate object. Then it is not ready-to-hand anymore but *present-at-hand* ('Vorhanden'): open to questions about what is the matter with it. We then have to reconsider our engagement with the object, repair it, replace it, or change our goal altogether.

Though computer icons cannot break down like a hammer, they may function as ready-to-hand or present-at-hand. They are ready-to-hand tools when they yield the expected result. We then take their iconology for granted and need not think about them as tools or as signs; we just click and go. But when a click fails, the icon is no longer ready-to-hand but present-at-hand, raising questions about whether the equipment chain is broken somewhere, or whether we just misinterpreted the sign. In Peircian terms: this raises questions about its reference to the object, which seemed to be unproblematically immediate when ready-to-hand, but exposes its dynamical and virtual mediations when suddenly becoming present-at-hand.⁴ From this perspective the transformation of the computer icon from an integrated ready-to-hand sign-tool into a decomposed present-at-hand object is not a disturbing inconvenience, but an opportunity. An opportunity to investigate how digital equipment works, how it refers to other equipment and code, how it conceals its indexical dynamics, how it can fool you, and how you can counter this. As Winograd and Flores (1985) put it: 'Breakdowns serve an extremely important cognitive function, revealing to us the nature of our practices and equipment, making them present-to-hand to us, perhaps for the first time. In this sense they function in a positive rather than negative way.'

Conceptual metaphors

My friend, looking for her mail, would probably have found little comfort in Heidegger's ready-to-hand and present-at-hand tools, or Peirce's immediate and dynamical objects. We are all seduced by the immediate object of the sign and tend to remain trapped in its iconology. We take computer icons as condensed sign-tools, without being aware of the constitutive parts. While this can partly be explained by what I have called iconology, this does not explain why, even in a situation of being stuck with a present-at-hand sign-tool, the sign and the tool part still tend to stick together. What kind of glue keeps them together? For an answer to this question, we have to zoom in on the mechanism of metaphorical substitution encountered earlier.

The shortest definition of metaphor is: something which stands for something else. While all signs in fact stand for something else, metaphors are special, since they do so in a double sense. Metaphorical signs condense two references by transferring and incorporating qualities from one object of reference into another. Hence, metaphors are mechanisms of transference (from one to another) and condensation (one within another). This transference can be accomplished by words, but equally so by images (Forceville 2008) and physical objects (Guten-

plan 2005). By assigning concepts to words, images and material objects, these ontologically different things are able to blend conceptually.

According to the so-called conceptual theory of metaphor (Lakoff and Johnson 1980; Lakoff 1993), metaphors can be analyzed as sets of conceptual correspondences across semantic domains, with an underlying basic conceptual metaphor which can be abbreviated as Y IS X, or TARGET DOMAIN IS SOURCE DOMAIN (Lakoff 1993). Might this be useful to analyze our riddle of the sign-tool iconology?

Analyzing the mailbox icon as a metaphor would yield E-MAIL IS POSTAL MAIL as basic conceptual metaphor, and the cross-domain mapping would look like Table 1.

| SOURCE DOMAIN: postal mail | TARGET DOMAIN: e-mail |
|------------------------------|--|
| mailbox | inbox of mail program |
| letters, packets | messages, attachments |
| sending and receiving | send or get mail button |
| sorting, disposing | distribution to folders, deleting |
| [postal distribution system] | [mail-server network at ISPs] |
| [delivery by postman] | [consulting a mail server; fetch mail command] |

Table 1: Cross-domain mapping for the conceptual metaphor E-MAIL IS POSTAL MAIL

All elements match well, except the last two in the list. These entities are indeed involved in the conceptual transference, but they are also exactly what is hidden in the iconic metaphor. They *are* and *are not* part of the conceptual mapping; they are what I have called *depresented*, while the other correspondences are represented as such on the interface. Note that this representation does not pertain to just unused parts of the conceptual domains. Unused parts would be ‘stamps’ in the source domain and ‘viruses’ in the target domain – these associations are simply ignored, they have no cross-domain correspondence, and they are not necessary to establish the metaphor of e-mail. But the aspects ‘postal distribution system’ and ‘delivery by postman’ in the source domain do have a counterpart in the target domain – they are indispensable for a successful transference. E-mail is only mail when it is also distributed and delivered. Yet, these acts are not represented in the metaphor connecting/condensing the source and target domain. Lakoff and Johnson’s model of conceptual metaphor thus gives an account of how the mailbox icon connects postal mail with e-mail on the level of the interface and human conceptual cognition, but it ignores the connections to software and machinery. It explains the human code, but not the machine code: the indexical, dynamical operations involved, such as a connection with an Internet service provider, a mail client consulting a mail server, and the subsequent transport of mail to the user’s PC.

Schematized (see Figure 1) this model of metaphor consists of an input (source domain) and an output (target domain), mediated by a cognitive human mind which is able to read the metaphorical mailbox icon as immediate object, thus connecting the source and the target. The necessary machine mediations – indexical references to dynamical objects – are invisible, hidden in a black box of software and machinery. Without a properly functioning dynamical black box, the output of the virtual object ('my e-mail') will not come about.

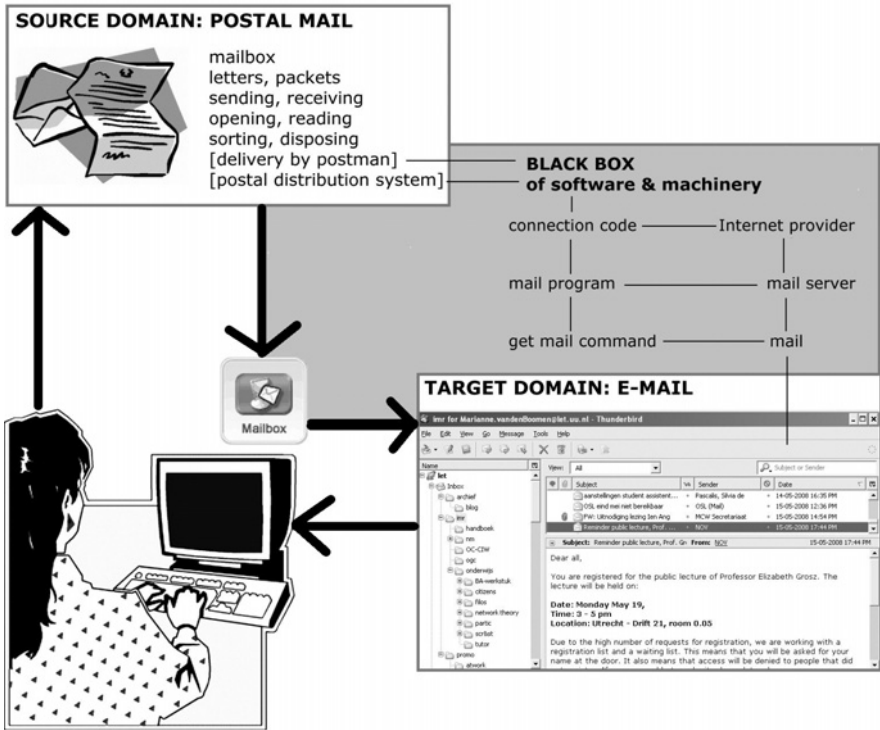


Figure 1: Input-output mechanism of the conceptual metaphor E-MAIL IS POSTAL MAIL

What is ignored in this model is that the metaphorical icon needs not only human reading but most of all action to invoke the transference – not only action by the user (configuring, clicking) but especially action inside the machine and the network. Meanwhile, the user is only able to read the icon as far as this action is represented and blackboxed. As we have seen, that is what the icon does by 'iconologizing': equating the sign with its immediate object, effacing its material indexical relations to dynamical objects, pretending it stands immediately for the virtual object in the target domain.

We may conclude that the classic, conceptual theory of metaphor in fact mirrors this iconology, as it focuses primarily on the interfacial relation with the immediate object, the object as represented/metaphorized by the sign. Though this underscores once more how a sign can conceal and supersede its tool-being, it still provides no proper explanation for how the compound of sign and tool gets glued together. What we thus need, on top of the theory of conceptual metaphor, is a theory of the processual materiality of metaphor. To my knowledge, there is no such theory yet. In the last section I will tentatively gather some possible building blocks for such a theory.

Material metaphor

What is needed is a theory of metaphor which not only maps transferences between different semantic domains – from concept to concept, as does the conceptual theory – but also between different ontological domains: from concepts to objects, from software instructions to visual icons, from signs to tools. Such a theory should provide an account of the involved material indexicality of sign-tools, of their relation to the dynamical objects inside the black box, and the relation with the hoped-for virtual object.

Katherine Hayles's (2002) notion of material metaphor is a promising starting point. Hayles defined material metaphor as the instance of metaphor where the metaphorical transference does not take place between semantic concepts, but between symbolic signs and material apparatus. Like all metaphors, material metaphors condense two references, yet they do not blend two conceptual references; rather, they blend a conceptual reference with a material indexical reference. Hayles proposed the term in the context of her medium-specific analysis of electronic literature. In this field written words can do more than just signify or represent – they may, as material metaphors, be able to perform acts: dance, move, dissolve, mutate, tear apart, destroy previously read words, etc., according to the underlying software code.

This kind of e-textual performance can be seen as a radicalization of Austin's speech act theory. Austin (1962) pointed out how verbal statements such as 'I promise' or 'I baptise this ship' are not just constative utterances but performative speech acts. By being uttered (in a specific context), such utterances *do things in the world*, they accomplish a change in a state of affairs. In a similar vein e-text is able to perform 'digital speech acts'. Though Austin's speech act theory pertains primarily to language and Hayles's material metaphor primarily to digital literature, there seems to be no reason to confine material metaphor to the domain of words and texts. Just as images and physical objects may be invoked as conceptual metaphors, they may be invoked as material metaphors. Whenever an image or an object functions as a symbol and is able to entail a change of affairs when enacted in the proper context, it can be considered a material metaphor. Take, for

example, the planting or capturing of a flag – this not only signifies colonization or victory, it evokes it by the very act of planting or capturing.

Material metaphors are ubiquitous in ordinary computer interfaces. They exist as sign-tools, as icons and hyperlinks, as pixelated buttons and switches which can indeed be switched on and off. Such material metaphors condense iconicity and indexicality by blending immediate and dynamical objects into one sign-tool. The mailbox icon is a material metaphor, able to invoke indexical material acts, changes in the material machinery of the computer and the network. Conceiving the mailbox as material metaphor thus reveals what is derepresented in the conceptual metaphor. While the conceptual metaphor can explain what happens inside the head of the user, the material metaphor leads us to the hands of user, and from there to and through the interface, right inside the black box of digital machinery.

Not that the black box will become transparent. Most parts of this black box will remain in the dark, since we humans are practically blind in this digital domain. We have delegated the relevant intentional acts to code, which we can execute but which we cannot grasp directly. We are simply not equipped with a perceptual or cognitive apparatus to read the digital, to read patterns of numbers and infer their meaning or effects. However, we do have partial, mediated access to the digital. We have this possibility because machinic calculations and translations can be ontologized and represented to us as readable signs, that is as material metaphors. We are able to read and act upon parts of the code, as far as these are represented as a sign-tool. This may be done by icons, but also by textual configuration menus ('mail server settings'), hyperlinks or other metaphorized commands ('get mail', 'send mail').

While these material metaphors mostly go by unnoticed, ready-to-hand and iconologized, they are however 'near at hand'. An analysis of both their conceptual iconicity and their material indexicality may turn them into keys to the black box, able to close and conceal, but also able to open and reveal. A full-blown theory of material metaphor may thus provide analytical apertures in the digital black box, enabling us to peek inside.

In order to develop such a theory, a very heterogeneous set of analytical tools and concepts should be assembled. Though Peirce, Heidegger, Lakoff, and Hayles may be strange bedfellows, I consider their joint forces necessary for this endeavor. The ongoing proliferation of cross- and intermediations induced by digitality results in more and more material-semiotic assemblages in the form of iconologized virtual objects. They could and should be hacked analytically.

Notes

1. For example Sherry Turkle's metaphorical book title *Life on the screen: Identity in the age of the Internet* (1995). In Steven Johnson's *Interface culture* (1997) the computer interface

- equals the screen with its visual metaphors. Even in Bolter and Grusin's *Remediation* (1999) the basic principles of mediation – immediacy and hypermediacy – are both defined as 'styles of visual representation' (272). See for other examples of screenic hegemony the chapters by Nanna Verhoeff and Ann-Sophie Lehmann. See for a further deconstruction of the notion of indexicality the chapter by Frank Kessler.
2. Average Internet users nowadays would perhaps not even understand the telephone metaphor as network connection icon, not being familiar anymore with dial-up Internet accounts. A telephone icon nowadays usually stands for another application: voice over IP.
 3. Cf. Saussure's structuralist semiology, in which only two ingredients of the sign are distinguished: signifier (material embodiment of the sign) and signified (mental concept, produced by the differences in the chain of signifiers, arbitrarily connected to a signifier). In Saussure's linguistic conception of the sign, any notion of reference to something outside language is disabled. While this precludes the pitfalls of claiming essentialist, natural correspondences between words and things, it also excludes any conceptualization of possible relations to a world outside human language.
 4. Cf. Latour's distinction between intermediaries, which just transmit, and mediators, which transform, translate and modify meaning while processing: 'A properly functioning computer could be taken as a good case of a complicated intermediary (...) But if it breaks down, a computer may turn into a horrendously complex mediator' (Latour 2005, 39).

References

- Austin, John Langshaw. 1962. *How to do things with words*, eds. J.O. Urmson and Marina Sbisa. Cambridge, MA: Harvard University Press.
- Bolter, Jay David, and Richard Grusin. 2000. *Remediation: Understanding new media*. Cambridge, MA: MIT Press.
- Coward, Rosalind, and John Ellis. 1977. *Language and materialism: Developments in semiology and the theory of the subject*. Boston: Routledge & Kegan Paul.
- Forceville, Charles. 2008. Metaphor in pictures and multimodal representations. In *The Cambridge handbook of metaphor and thought*, ed. Raymond Gibbs. Cambridge, MA: Cambridge University Press.
- Freadman, Anne. 2004. *The machinery of talk: Charles Peirce and the sign hypothesis*. Stanford: Stanford University Press.
- Gutenplan, Samuel. 2005. *Objects of metaphor*. Oxford: Claridon Press.
- Harman, Graham. 2002. *Tool-Being: Heidegger and the metaphysics of objects*. Chicago: Open Court Publishing.
- Hayles, N. Katherine. 1993. Virtual bodies and flickering signifiers. *October* (66): 69-91.
- . 2002. *Writing machines*. Cambridge, MA: MIT Press.
- . 2003. Translating media: Why we should rethink textuality. *The Yale Journal of Criticism* 16 (2): 263-290.
- . 2005. *My mother was a computer*. Chicago: University of Chicago Press.
- Heidegger, Martin. 1954. The question concerning technology. In *The question concerning technology and other essays, 1977*, ed. William Lovitt, 3-35. Harper Torchbooks.
- Heidegger, Martin. 1927. *Sein und Zeit*. Tübingen: Max Niemeyer Verlag.

- Johnson, Steven. 1997. *Interface culture: How new technology transforms the way we create and communicate*. New York: Basic Books.
- Lakoff, George. 1993. The contemporary theory of metaphor. In *Metaphor and thought*, ed. Andrew Ortony, 202-251. Cambridge, MA: Cambridge University Press.
- Lakoff, George, and Mark Johnson. 1980. *Metaphors we live by*. Chicago: University of Chicago Press.
- Latour, Bruno. 2005. *Reassembling the social: An introduction to actor-network-theory*. Oxford: Oxford University Press.
- Lucy, Niall. 2001. *Beyond semiotics: Text, culture and technology*. London, New York: Continuum.
- Manovich, Lev. 2001. *The language of new media*. Cambridge, MA: MIT Press.
- Peirce, Charles Sanders. 1873. Of logic as a study of signs. In *Writings of Charles S. Peirce*, 1981, ed. Peirce Edition Project, Vol. 3.82, Bloomington: Indiana University Press.
- Peirce, Charles Sanders. 1888. A guess at the riddle. In *Collected papers of Charles Sanders Peirce*, 1932, eds. Charles Hartshorne, Paul Weiss, and Arthur W. Burks, 365-367. Cambridge, MA: Harvard University Press.
- Peirce, Charles Sanders. 1904. A letter to Lady Welby. In *Collected Papers of Charles Peirce*, 1958, ed. A.W. Burks, Vol. 8.328. Cambridge, MA: Belknap Press of Harvard University Press.
- Peirce, Charles Sanders. 1909. A letter to William James. In *The Essential Peirce: Selected philosophical writings*, 1998, ed. Peirce Edition Project, Vol. 2.498. Bloomington, Indianapolis: Indiana University Press.
- Saussure, de, Ferdinand. 1983 [1916]. *Course in general linguistics*, eds. Charles Bally and Albert Sechehaye. La Salle, IL: Open Court Publishing.
- Tilley, Christopher. 1999. *Metaphor and material culture*. Oxford: Blackwell Publishers.
- Turkle, Sherry. 1995. *Life on the screen: Identity in the age of the Internet*. New York: Simon & Schuster.
- Winograd, Terry, and Fernando Flores, eds. 1985. *Understanding computers and cognition*. Norwood: Ablex Publishing.