

Paris Utrecht Mini Symposium 'Meta'
Paris 19 May 2006

[1] Metaphors of the metasphere

During this conference we already have seen a lot of different notions of meta, and I will add yet another meta thing.

I will talk about metaphors. My research is about how metaphors mediate our interaction with digital technology, in particular the Internet. You might think of metaphors such as the electronic highway or the global village, but I am especially interested in concepts which might not be immediately recognized as metaphors.

[2] As for instance [Web 2.0](#).

Indeed, this is a metaphor – a conceptual metaphor as defined by Lakoff and Johnson: i.e. a concept taken from a source domain and transferred to a target domain, in order to say something in terms of another thing.

The signified target domain of Web 2.0 is not very clear, that's of course the very reason a metaphor is mobilized. This vague [target domain](#) consists of a loose collection of social and technological qualities, such as:

- freely available webbased applications instead of personally installed software
- user input and participation to enlarge and enrich the collected data (a.k.a. 'collective intelligence')
- the use of scripting technologies, preferably Ajax
- and though Web 2.0 need not necessarily be commercial, it bears the promise of a new business model.

That's a lot of things to articulate at once, so the help of a short cut, a metaphor, is called in. Where does the notion Web 2.0 come from? What is the conceptual [source domain](#)?

This clearly pertains to the release of software packages, where it is usance to number subsequent versions.

A release of version 2.0 in the software field implies:

- a new release after an older version 1.0
- a release which is not only patched but fundamentally improved
- an integral package, released at a specific moment in time
- the urge is to update/upgrade, or you will be lagging behind
- and though a new software release need not necessarily be commercial, it certainly has a shade of branding and marketing.

[3] What I like about this metaphor is that it focuses on software as such. Most metaphors to indicate something digital are leading our attention away from software; this one foregrounds it. That's special.

This is all the more important since I am convinced that [metaphors provide our only access to the](#)

digital. After all, the digital consists of sets of ones and zero's, only readable by machines, not by humans. Humans need metaphorical translations in order to understand, operate and control the digital. Yet, most metaphors do this by concealing software processes. Think of a metaphor such as 'collective intelligence'.

So indeed it is epistemologically empowering to have a metaphor as Web 2.0, a metaphor which keeps an open eye on software as such, on it's existence as code.

But, as all metaphors, the metaphor of Web 2.0 both **reveals and conceals**. It highlights particular qualities, and it downplays other qualities. What is highlighted in Web 2.0 is the total **revolutionary break**, and most of all: the suggestion of a coherent **unified package**. Revolution from the shelf, we might say.

What is downplayed is that the phenomena in the target domain are not a unified whole, but complex **distributed cooperations between heterogeneous technical and social actors**. These actors are for instance: network protocols, clients and servers, programs, scripts, databases, hyperlinks, tags, user content, user data, and user semantics.

The cooperations between those actors can fail or succeed, and anyway, a lot of trial and error is involved. In that sense the software metaphor of the 'beta version' might be more appropriate.

This does not imply I want to dismiss the metaphor Web 2.0 all together. On the contrary, I think any sticky metaphor of digital dynamics should be taken seriously, since they articulate particular access to the digital.

Taking a metaphor seriously also means taking it literally. So let's take the metaphor Web 2.0 literal, and see where it leads us.

[4] It leads us to the history and development of the Web, **starting at Web 1.0**.

What is Web 1.0?

In the mid-nineties of the last century the World Wide Web came up as as a new application. It was, as most applications on the Internet, a client-server application, and it used its own http-protocol to transfer files. A web of linked pages emerged as a new extendible **information platform**.

New was the **multi-media** aspect: the HTML-assemblage of text, graphics, and images, delivered as one integral page. The components constituting the page could reside on one webserver, but also on several dispersed webserver. Relatively new was the easy **hyperlinking** to other sites.

Dynamic as this may look, webpages at that time just supported reading and clicking – nothing more. Communication was exclusively reserved to non-web applications such as e-mail, Usenet and IRC. On the Web users could at most fill in basic forms, but these were handled by hidden e-mail. Anyway, users could never change or add something on a webpage. In short, webpages at that time were **static**.

Moreover, while the 1995 hype claimed 'everyone can now be a publisher' websites had to be maintained by skilled **webmasters**. Only they could decide whether to add a new page or hyperlink.

A few things changed with the advent of so called **scripts**. Let's call this moment Web 1.1, to stay

within the metaphor.

[5] Web 1.1: the scripted web

What is Web 1.1? An update towards a more dynamic web by introducing **scripts**. Scripts are small routine programs which can perform actions on data components. They consist of executable source code, as an extra software layer on top of the web protocol.

These scripts finally enabled **user interactivity** on a web page. Scripts emerged for polls, quizzes, banner delivery, search forms, database queries and member's access rights. But most important were the scripts which simulated **communication** applications such as Usenet and IRC. This resulted in webforums and webchatrooms. These scripts usually work with a database running in the background – a new actor on the web. Scripted websites could not only assemble and execute plain html-, image-, and movie-files, but also script-files and database entries.

The implementation of scripts was still in the hands of the webmasters – they now also had to manage databases and scripts – but from now on webpages could be **both an information and a communication platform**. The scripted Web introduced **dynamics**. Pages could be transformed by user input – by adding user content, or by retrieving a personalized page from a database.

Now the Web really could take off. And it did.

[6] Though it is tempting to see the scripted web as a technical revolution, it was already implied the **basic principles of digitality**, as formulated by Lev Manovich in his classic book *The language of new media*. These principles are:

1. **numerical representation** – i.e. the assignment of numbers to discrete elements. Numbers are assigned to data objects such as pixels, letters, images, files and database tables. And they are assigned to algorithmic entities residing in operating systems, programs, protocols and scripts. In fact – software, code, is basically this: assigning numbers to discrete elements, so they become computable. And computable not only means calculable but also: able to change hardware states of a machine.
2. **modularity** – i.e. the possibility of defining independent sets of numbers, thereby creating independent modules which can work together or be assembled. Again this can be done on the data level (such as files, images, messages, database entries) but also on the algorithmic level: patches, plug ins, scripts, viruses.
3. **variation** – which pertains at the fact that each data object is computable and therefore can be edited, filtered or otherwise manipulated. Software and data objects are principally mutable, extendable and hackable.
4. **automation** – the computational variations can be invoked by human actors, but they also can be automated by software. Again, this can be done on all modular levels: operating systems, protocols, programs, scripts, macros and viruses. And each of these can be taken up in a chain of

programmable causation between modules.

5. Manovich' fifth principle is a special one, because this one is not purely based on software ontology. He calls it [transcoding](#).

This refers to the mutual exchange between cultural forms and computable forms.

Transcoding is the moment of translation between machine readable computational code and human readable cultural code, and vice versa. The ontological gap between the cultural code and the computer code is bridged and mediated by several boundary objects. These objects may reside as icons or textual menu's on our computer interface, but they also exist more fluid as conventions, concepts, and analogies. These forms circulate in our cultural world and can be transferred to get conceptual access to the digital domain.

This is where metaphors come into play. Again, think of transcoding metaphors such as the electronic highway, but also of metaphors such as 'page', 'desk top' or 'virtual community'.

[7] Before zooming in on the metaphor of community, I want to [add a 6th principle to Manovich' list](#). Because, remarkably, none of Manovich' principles gives an account of the possibility of networked computing. None of the five principles contains a key to the fact that all principles can work at distance, can be enacted on geographically and physically different computer systems.

This enactment is only possible because of a another ontological feature of software:

6. [mobility](#).

This mobility is based on the numerical representation in binary digits, resulting in patterns of bits, ones and zero's, which on its turn can be represented, stored and transported. This can be done by any medium which is able to preserve the differential pattern.

The Internet would not exist without this principle of mobility. Sending and receiving mail, downloading files, browsing web pages – these acts all rely on the mobility of data patterns. Internet mobility is regulated by the basic [TCP/IP protocol](#), which manages the slicing of data packets, the transport over the network and the reassembling at their final destination. On top of this basic transmission protocol every [application](#) (such as mail, usenet, irc, web) has its own [protocol](#), in order to process and reassemble their own modular entities.

Now here something interesting takes place. Applications represent their protocologically reassembled modular entities as belonging together. This can take the form of listed messages in usenet newsgroups, dialogues on a IRC–channel, or listed forums on a webpage.

Though these are screen representations, they are conceived by users as [spaces](#). Spaces where you can go to, where you can be. You *go* to a webpage or a webforum, to a usenet group or an IRC channel.

This is transcoding at work. Seeing a screen representation as a space is translating a computer form into a cultural form. And it does not stop here. When user communication is possible in these spaces,

they can be become spaces of social gathering, public spaces.

[8] That is where the transcoding **metaphor of virtual community** comes in, a metaphor to indicate social and cultural exchange in virtual spaces.

The **target domain** of this metaphor can be defined as a localized enduring **social aggregation** on the Internet, having a core of **frequent users**, thriving on ongoing group **communication** and exchange, and developing it's own **shared values and morality**. Members might develop personal relationships and meet face to face, but not necessarily. Virtual communities might directly relate to real life practices, but not necessarily.

The **source domain** of the metaphor is a **pre-modern idyllic village** (take notice that the concept came up in 19th century modernity, driven by anxiety about the erosion of traditional communities by industrial society). This village had **clear borders**, an almost **self-sufficient** economy, and a strong **social cohesion**. Everyone knew his place, his duties and one another.

In the early nineties the metaphor of community served to highlight the communicative and social bonding capacities of computers. Which was indeed necessary, since at that time computers were mainly conceived als cold calculation machines. The concept of community was also associated with idealism and bottom up democracy, but nowadays this utopian stance is dissolved. What remained is the association with a **bordered space and a core group of recurrent users**.

It should be noted that the early models of virtual communities were located on bulletin board systems, mailinglists, Usenet groups, IRC channels and textual role playing environments – not on the Web. Web 1.0 was a non-place for community building, since it provided no space for social gathering and communication. This changed with the emergence of the scripted web, which enabled the formation of webcommunities on fora and chatrooms.

So back to our scripted web, Web 1.1.

[9] Web 1.1 to Web 1.2

In Web 1.1 virtual communities could emerge, since group communication was now possible. This invoked a booming of community portals on the web, including new business models to exploit it.

Yet, what made these webcommunities **less dynamic** than usenet groups, was the fact that the group communication remained internal, **within the borders of the webpage**. Where mailing lists and Usenet groups could crosspost messages to other lists and groups, thereby mingling and extending their debates, webcommunity debates were confined to their specific forum or chatroom.

Hyperlinking of course was possible, but these links were just **one-way information pointers**, not two way connection and communication channels.

This changed with the **emergence of blogs**. I propose to call this **Web 1.2**.

What is a blog?

Most definitions focus on the [diary and logbook format](#), i.e. the format of a page, consisting of a list of [separate entries](#) stamped by date, the latest entry on top of the page.

What makes the blog more than an information platform is that every entry might function as a [separate webforum](#), by allowing comments and links from readers. A blog is maintained by one person, the blogowner, or by a group of blogowners. It might also be collective, when both comments and main entries can be submitted by ordinary users.

So a blog is a hybrid of information and communication entries, and it blurs the strict labour division between webmasters and webusers.

[10] But there is more. Blogs embody a [radicalization of the scripted web](#). On the software level a blog is nothing more than a webbased content management system, consisting of a set of scripts and a database. The blog scripts might be [server side scripts](#) (to be managed by a webmaster/blogowner) or they might be [client-side](#) scripts – delivered as free webservice to ordinary users, who only need their webbrowser to start a blog. Now indeed 'everyone can be a publisher'.

Again, the radicalization of the scripted web works according to the six principles of software ontology.

Especially the [modularity](#) and [mobility](#) of data objects is pushed to the edge. Data objects on a blogpage can be reassembled from the blog database in several ways: by subsequent [date](#), by monthly [archives](#), by full text [search](#) words, by blogowner defined [categories](#), and on some blogs even by [user evaluations](#). When blog entries are [tagged](#) by the blogowner, they can be reassembled according to their tag, together with identically tagged entries on other blogs. Thereby not only reassembling internal blog objects, but reassembling a web of blogs.

We might say: blogs [radicalized the hyperlink](#).

While Web 1.0 hyperlinks could only refer to a complete webpage, bloglinks are far more [calibrated](#), since they can refer to parts of a page.

So called [permalinks](#) refer to separate entries and separate comments, both within a blog and between blogs. So called [trackbacks](#) are even more interesting, since they function as crossposting and two way hyperlinks. A trackback is a notification appearing on your blog that another blog has referred to an entry of yours. The notification consists of a short summary and a link to the referring blog entry. This allows for conversations spanning several blogs that readers can easily follow.

So, what blogs do with all kinds of extra layers of scripts is basically cutting webpages into separate pieces and enabling several ways of reassembling them. Not only on a particular blog, but also between blogs, in the so called [blogosphere](#), the web of hyperlinked blogs.

Whether you call this revolutionary or not, Web 2.0 or Web 1.2, this is a relatively new web dynamics, based on aggregating and reassembling data components [within and between websites and databases](#). The HTML page itself is no longer a confined unity but consists of heterogeneous data

parts, which can be addressed and processed separately. The age of the page is over.

[11] This means [trouble for virtual communities](#), since they are traditionally grounded in [bordered protocological spaces](#).

Protocols regulate specific ways of transmitting, and thereby create network topologies and virtual spaces. This architecture enables community formation, by facilitating users to produce their own content and communication styles within these spaces.

The scripted web can be analyzed as a [meta-layer](#) on top of these protocols. It does parasitize on protocological spaces, but its primary focus is calibrated content. Its primary focus is on gathering, extracting, selecting, distributing and reassembling data objects.

On the fly scripted spaces might be created – of course, a blog may assemble a core group of recurring contributors, thereby constituting a community. But the social bonding power of blogs lies more in its data gathering than in its people gathering. More in its ability to mobilize and distribute [user data input](#) than to mobilize [users-as-such](#).

In that sense, blogs, and other scripted webparasites hyped as Web 2.0, are radically decentering technologies. They even decenter virtual space and virtual subjects, by thriving on distributed user data input.

In short, blogs, with their distributed data spaces, disrupt the classic notion of virtual community. I argue that we should refrain from the community metaphor when trying to grasp the distributed dynamics of the scripted web.

Do we have other terms to signify and demarcate particular clusters in this distributed soup? I am not sure.

Concepts as ['websphere'](#), ['tagclouds'](#) and ['blogosphere'](#) are persistent circulating memes, they might contain keys to insightful new metaphors. After all, these concepts point in the same direction: a gaslike volatile sphere of assembling and reassembling distributed data instead of a solid space which assembles people.

Perhaps scripting can be analysed as producing [metaspheres](#). In these metaspheres the social does not consist of communities of like minded human beings, but of parasitical meta-assemblages of heterogeneous actors: nested protocols, layered scripts, distributed data, and distributed user's acts. But again, I am not sure.