Bernhard Rieder, Université Paris 8

bernhard.rieder@univ-paris8.fr – http://procspace.net/berno/

Mirko Tobias Schäfer, Universiteit Utrecht

mirko.schaefer@let.uu.nl – http://dvision.at/mirko/

Beyond Engineering - Software Design as Bridge over the Culture / Technology Dichotomy

In his book "Le geste et la parole", the paleontologist André Leroi-Gourhan sketched the evolution of *Homo sapiens* as having left the domain of biological advancement in order to continue – with accelerated pace – in the field of language and technology. While many of Leroi-Gourhan's proposals have not aged well, his concept of the modern world as a dynamic web of objects and symbols – of machinery and discourse one might say – is a powerful image in a time where the idea of the tool as neutral artifact is still the dominant paradigm. In the last decades, however, there has been growing interest in technology not as means to ends but as a cultural force. Together with this shift in perspective on the role of technical artifacts in our high-tech societies, we see, more specifically, an increased awareness for the "toolmaker" as the supposed *locus* of technical progress. Every age seems to have an epitomical figure of the technical creator: the craftsman for the Middle Ages, the inventor for the Industrial Revolution, and the engineer for the 20th century. In late capitalism, there is a new figure appearing on the horizon: the designer as the toolmaker of the information age.

The last two decades have produced a plethora of literature on the new way of creating tools and objects; from product design to Web design, from graphic design to experience design, design is everywhere and – of course – no two definitions are the same. As a consequence, the term refers less to a clear-cut concept or methodology; it rather functions as a means of differentiation. Software design[1] for example is not a well-defined practice; it is a way of saying that what is being done is somehow going beyond the well-defined practice of software engineering. Behind the term "design" actually lurks a multiplicity of (new) ways of creating, shaping, and maybe even using.

[1] The text that first launched the debate was Kapor, Mitchell: A Software Design Manifesto. In: Winograd, Terry: Bringing Design to Software. New York: ACM Press, 1996 pp.1-9 [first published 1990] (http://hci.stanford.edu/bds/1-kapor.html)

In this article, will first look at how software is shaping culture; we will then take up the area of software production to show how the new practices are connected to and stimulated by a still strange artifact, the Universal Machine, in order to arrive at the core of our argument: if culture and technology are increasingly difficult to distinguish, we will have to take a second look at how we *create* tools, *think* about culture, and *regulate* society.

1. Hybrid Practices

Since the advent of digital computers in the late forties and especially the marketing of the consumer PC in the eighties, information technology (IT) has grown to be ubiquitous. But while "computer" and "technology" have almost become synonyms and the basic technical principle has stayed the same for the last sixty years, there remains an aura of vagueness around these machines. Herein actually lays their power: computers themselves are *functionally underdetermined*, they need software to turn them into complete devices with distinct functions. While the hardware is the necessary base layer, the Universal Machine, the *specific* machine – a series of functions and procedures that manipulate information and, with proper connection, matter and energy – is the result of programming. In the words of Alan Turing,

> "The importance of the universal machine is clear. We do not need to have an infinity of different machines in doing different jobs. A single one will suffice. The engineering problem of producing various machines for various jobs is replaced by the office work of 'programming' the universal machine to do these jobs."[2]

These words mark not only the technical novelty but also one reason for the cultural significance of IT: somebody who buys a computer today does not only get the *physical* machinery, but also gains access to a seemingly indefinite world of *logical* machinery. These software programs spring from a fecund environment of different work styles that nowadays go well beyond the classical methods of engineering or even beyond the "office work" Turing mentions. We will come back to this question in the second chapter of this paper.

In industrial societies, the professional world has few tasks left that are not in one way or another dependent on computers – and the private sphere seems to follow suit. Our communication and information habits have shifted in large parts to computer-based networks and with the advent of VoIP, 3G telephony, and TV over ADSL, most of the classical electronic media will have passed onto

[2] Turing, Alan M.: Intelligent Machinery. National Physical Laboratory Report, 1948 (http://www.alanturing.net/turing_archive/archive/l/l32/L32-001.html)

the universal protocol of TCP/IP, becoming yet another piece of software that runs on a network of globally connected Universal Machines, the metamedia[3] of our time. Creative work, game play, social intercourse, information search and management – so many of the things we do in our everyday lives have become directly connected to digital tools and networks; and there is no end in sight.[4]

The Humanities have historically shown moderate interest in technology. Most thinkers assimilated technical artifacts into the industrial complex and treated them as producers of capital rather than meaning – with the notable exception of McLuhan's poetic formalism. In recent years, with IT being integrated into the very fabric of cultural practices, we have started to look closer at how technology shapes the human condition.

What we need today is not a grand theory of how technology and society relate on a general level, but a perspective that zooms in on the micro-level of entanglement and theorizes not only the great developments, but first and foremost the everyday practices[5] that are the content of human affairs. With Michel de Certeau, we can describe a practice as an "art de faire" – a way of doing that embeds an action in a dense network of meaning, providing a reason for why something is done and a proper way of doing it. There is obviously a non-discursive dimension to a practice (the pure motor movement), as well as a strong discursive element (morals, laws, rules, narratives, etc.), woven together by continuous action. These atomic particles of practices, actions, have been theorized by Harry Collins and Martin Kusch[6] as trees made of micro-acts, with motor movement at the bottom and meaning on the top. The action "talking to somebody" actually contains a variety of acts that go from the ritualized "greeting" and "waiting for your turn to speak" at the top, down to the forming of sounds in your mouth. There is no neat separation between the two, but a progressive chain linking top to bottom and the other way round, with a large zone of intermingling in the middle. If we understand practice as an embedding of action in time and habit, the discursive dimension can no longer be severed from its non-discursive counterpart.

---

[3] Kay, Alan / Goldberg, Adele: Personal Dynamic Media. In: The New Media Reader. Cambridge MA: MIT Press, 2003 [first published 1977] pp. 393-404

[4] "The networking logic epitomized by the Internet became applicable to every domain of activity, to every context, and to every location that could be electronically connected." Castells, Manuel: The Information Age: Economy, Society and Culture. Volume 1: The Rise of the Network Society. Malden MA: Blackwell, 2000 [first published 1996] p.52

[5] De Certeau, Michel: L'invention du quotidien. Paris: Gallimard, 1994 [first published 1980]

[6] Collins, Harry / Kusch, Martin: The Shape of Actions. What Humans and Machines Can Do. Cambridge MA, London: MIT Press, 1998

When applying this view, we see that what is happening with technology in general and with IT in accelerated and enlarged form, is that actions are becoming intrinsically hybrid because we delegate always larger parts of the action tree to machines: talking to somebody now very often implies micro-acts performed by a machine such as transporting a voice signal from A to B. As a consequence, our practices have become riddled with the work of machines, in many cases without us even noticing. Many of the tasks being delegated are "semantic" in nature: among other things, algorithms now filter, structure, interpret, and visualize information in an automatic fashion.[7] Through this dynamic, technology intimately intertwines with society on the micro-level – to a point where even the analytical separation of the two becomes highly problematic[8]. From a practical standpoint, we can summarize the vectors of hybridization in two classes:

- Actions and practices that are done differently (different in form, style, speed, efficiency, difficulty, range, etc.)
- Actions and practices which were not possible before (drawing on a virtual canvas, video communication across oceans, real-time data-mining, etc.)

While our distinction between the discursive and the non-discursive draws heavily on the work of Michel Foucault, there is one point where we have to go slightly further than he did: we believe that meaning is deeply embedded in the non-discursive, for example in software, and that in a sense technology is not only surrounded by discourse, but that it *is* discourse. We do not share Heidegger's hostile stance toward technology, but his understanding of the tool not as a means to some end, but as an ontological agent, as a way of "Entbergen", seems quite appealing to us. In "Gestell", the discursive and the non-discursive conflate; it is both object and logic – a "diagram", in the terms of Foucault, but with the difference in nature between the two planes largely gone. The lesson we take from this is diametrically opposed to Heidegger's position: involvement instead of withdrawal. We believe that technology is not one but multiple ways of revealing being and that the way we create technical artifacts – and software as the most important of such artifacts – is heavily influencing the cultural role they will play.

[7] Rieder, Bernhard: Processed Meaning. Perspectives on Semantic Computing in a Hybrid Culture. Ciberart Conference Bilbao, April 2004 (http://procspace.net/berno/files/semantic_computing.pdf)
[8] "The object that sits before the subject and the subject that faces the object are polemical entities, not innocent metaphysical inhabitants of the world." Latour, Bruno: Pandora's Hope. Essays on the Reality of Science Studies. Cambridge MA, London: Harvard University Press, 1999 p.294

2. Software, the Internet, and Design

In the passage cited above, Alan Turing not only offers a description of the Universal Machine, but by mentioning the "office work of programming", he already indicates that the computer implies changes in the way technology is fabricated. Today we see that the specific qualities of software and the global connection of users on the Internet have led to new configurations in cultural production. Before we can describe these new forms, we have to take a closer look at the technical elements of the equation.

2.1 Software

Software is *tentative*, always remaining "unfinished". In contrast to other technological artifacts, software development comes to an end when the program is working stable, not when it is "finished". While most industrial goods are perceived as products, software is best described as a process. The moment a program is released to the market it enters another stage of development, where code is progressively modified, fixed, and upgraded. Pierre Lévy describes this phenomenon as a permanent reading-and-writing continuum, rather a process than a product.[9]

Software is *modular* and therefore allows using parts of one program in the context of another one, which might go in a completely different direction – similar to sampling in DJ Culture. Since the functional range of software is hardly predictable, it can serve purposes different from what the programmer originally intended, and when assembled in chains, one program may use the results of another, thus creating functional amalgams where the final output is the outcome of an interaction between applications that "parasite" each other. Again because of modularity, complex software projects can be split into smaller parts that are developed separately and put together in the end. This way a collective work process is facilitated and a large group of programmers/designers can participate in a project by each providing only small modules that will make up the whole at the end.

9 Even before publication, the design process can be described as a reading-and-writing-continuum. Development methodologies such as the waterfall model, rapid prototyping, or the spiral model are based on reevaluation and maintenance (refactoring). See Lévy, Pierre: Collective Intelligence. New York: Plenum Trade, 1997 [first published 1994]

Software is *immaterial*, and in that aspect unlike other products of the cultural industries. It is similar to language[10] concerning structure and similar to machinery concerning functionality and effect. In some sense, when programming, the thought experiment becomes the experiment itself. We may even go as far as speaking of an apparatus of production specific to the creation of software. From an economic perspective, production and reproduction are configured differently from classic products: after purchasing a computer, the cost of producing and using (consuming) software is equal, time becoming the only limiting factor. This allows a great number of people to turn from consumer to producer. The basic "material" is code and the collective intelligence of users is constantly adding to available programs, effectively enlarging possibilities for design. Very much as with knowledge, the more people are adding to available ideas and models the more software becomes a cultural resource. This is possible because software is somewhere *between* words and things.

2.2 The Internet

The Internet distributes the qualities of software into space, making them available for everybody with an internet connection. Somebody who owns a computer not only gains access to the vast world of logical machinery, but also to a world of social interaction and collaborative work. Digital networks foster the construction of communities of interest and enable these communities to organize themselves and their (potentially global) workforce in an efficient manner. People are able to collaborate no matter where they are located or which time zone they are living in. Platforms such as Sourceforge.net host thousands of open source software projects and simultaneously provide the infrastructure for communicating between developers, for parallel programming, for debugging the software, for code maintenance, and for public representation to their work and themselves. For a dispersed programming community, the Internet becomes the media and the "memory" for creating, exercising, and representing their collective intelligence.

2.3 Design as Plural

Although the mentioned characteristics of software and the Internet play a significant role, there are important social and cultural dynamics at work in the current unfolding of software design as

[10] The analogy already gave the title to one of the major works on computers: Nelson, Theodor H.: Literary Machines 93.1. Sausalito, CA: Mindful Press, 1993 [first published 1980]

plural. According to IEEE Standard 610.12[11], software engineering is "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software". While there is of course a lot of software developing that approximates these guidelines, a huge fraction of the programming and design work strays very far from this definition. While this does not automatically make it "art"[12], we have to recognize that not a small part of the software we use has been created in the context of practices that go beyond the rigid requirements of software engineering. Computers and the Internet acted as enabling technologies that gave users the opportunity to extend the culture industry and to participate in cultural production[13], stimulating the social dynamic we witness today. Castells[14] recognizes the computer subcultures of the eighties as important roots, where techno-meritocratic values were more important than race, gender or economic status. In these communities, being good at writing code, breaking copy protections, and distributing information and knowledge was associated with "coolness" and respect. Even today, software programming is infused with values from youth culture, where fun, recognition by peers, experimentation, creativity, contestation, and the search for identity are the major factors of motivation. In contrast to engineering, software design often happens in a playful mindset more associated with leisure than with labor, fostered by the possibility – provided by the Internet – to work at home outside the schedule of office work hours.

The current evolution of software creation toward a plurality of "ways of doing" is also connected to questions of education and institution. In contrast to classical engineering, programming is often learned in an independent, autodidactic manner. The Internet offers the possibility to meet more experienced programmers who are willing to give a hand and there is an infinity of open code to learn from. The education of software designers is therefore increasingly detached from the classic institutions and has become at least co-organized in online communities. At the same time, knowledge production is not confined to research institutions and labs; many of the most innovative pieces of code appear seemingly without any connection to the established places of innovation.

---

[11] See: http://standards.ieee.org/catalog/olis/se.html

[12] For a discussion of the similarities between programming and artistic practice see Graham, Paul: Hackers and Painters. Lecture at Harvard, May 2003 (http://www.paulgraham.com/hp.html)

[13] These aspects were theorized lately by Jenkins, Henry: Interactive Audiences? In: Harries, Dan (ed.): The New Media Book. London: British Film Institute, 2002 (http://web.mit.edu/21fms/www/faculty/henry3/collective%20intelligence.html)

[14] Castells, op.cit., p. 39-40

These developments are not aimed at replacing the traditional and more organized institutions of work, education, and research; what we witness today is a trend toward enlargement and supplementation. With reference to an influential text written by Eric Raymond[15], we could say that the bazaar is not replacing the cathedral; it is blossoming in the city streets around it. The result is a plural of manners and settings in which software is created: from the teenager creating new levels for her favorite game to the obsessive programmer working on the intestines of a free operating system, the range of role models is continuously growing.

2.4 An Extended Culture Industry

We already see the emergence of hybrids of "old" and "new": corporate companies recognize the advantages of collective, non-hierarchic work processes and try to integrate the voluntary workforce of users into their production processes. The game industry for example receives a lot of input by user communities that modify their products. The commercial computer game Half Life for instance lacked a decent multiplayer mode; a few enthusiast gamers modified it to become the world's most successful multiplayer game, Counterstrike. The success of Counterstrike pushed the Half Life sales as well. By now it is commonplace to release a computer game with a level and content editor.

There is a major shift in cultural production currently taking place: alternative products, produced outside the established channels of the culture industry are being distributed globally at very little cost. In this aspect, the design bazaar is turning into serious competition to corporate companies. For the first time "consumers" are not only modifying products but are changing parts of the production apparatus[16] itself. We call this extension of production and distribution processes into homes and leisure the *extended culture industry*. The intertwined networks of cultural production that span between companies and users go beyond the mono-directional processes Adorno and Horkheimer were criticizing[17].

---

[15] Raymond, Eric S.: The Cathedral and the Bazaar. First Monday, Vol.3, Is.3, 1998 (http://www.firstmonday.org/issues/issue3_3/raymond/)

[16] Walter Benjamin calls for a shift from adapting the products of cultural industry to adapting the apparatus of production itself. This shift turns readers and viewers into participants. Benjamin, Walter: Der Autor als Produzent. In: Benjamin, Walter: Medienästhetische Schriften. Frankfurt a.M.: Suhrkamp, 2002 pp. 231-247 [first published 1934]

[17] Adorno, Theordor / Horkheimer, Max: Dialektik der Aufklärung. Frankfurt a.M.: Fischer, 1988 [first published 1944] The revelant chapter on cultural industry is online at http://www.marxists.org/reference/subject/philosophy/works/ge/adorno.htm

When tying up the different strains, the image of a significant shift in the production, distribution, and consumption in one of the most important sectors of technology emerges. While these developments are still difficult to seize, it is already possible to think about some consequences.

3. Bridging the Culture / Technology Divide

So far, we have made two separate arguments: first, we tried to show that software as logical machinery plays an important role in shaping our everyday life, accentuating culture as a hybrid of technology and discourse; and second, we stated that software has come to be developed in heterogeneous, multiple, and contradictory environments where creative practices flourish outside of classical institutions and methodology. In the third part of this paper, we want to briefly discuss these two arguments according to what they mean in three different contexts; the question is how the humanities, design discourse, and policy making are affected by technological creativity beyond engineering.

3.1 Humanities

Traditionally, philosophy and cultural theory have subscribed to a view of technology as something external to – or at least different from – society and culture. In this perspective, the practice of creating a technical artifact is very dissimilar in nature from processes of symbolization, e.g. the writing of law or literature. The first is supposedly oriented toward the material domination of our "lifeworld" (Lebenswelt) through efficiency while the second is concerned with the social (law) or cultural (literature) dimension of human existence. This separation has the convenient effect of exempting philosophers of technology of any need for technical knowledge because "technoscience" produces but more of the same, the true challenge lying in the discovery of the essential dynamics between the strata – endeavor reserved to the masters of symbolization. But there is a very dangerous side to this outlook: subtracting the dimension of *meaning* from technology implies subtracting *responsibility*. If the creation of technology is not understood to be a deeply cultural, social, symbolic, and political activity, there is no reason for the creators to adopt whatever ethical and political stance toward their work beyond the question of physical harm to others. We believe that in a time where logical machinery takes part in so many of the practices that make up our lives, we need a concept that is not only aware of "effects" of technology on

culture, of *technê* on *logos*, but which recognizes that technology *is* a form of culture, that *technê is logos*. Such a technical *logos* shares not the homogenous logic of "Gestell", but differentiates continuously into a plurality of forms and effects.

The closest approximation to our demand seems to us the "actor-network theory" championed by Bruno Latour and others. In this framework, the terms "technology" and "society" are both replaced by the concept of a *collective* of human and non-human actors. By abolishing this basic dichotomy, a path is opened that leads to the heterogeneous practices of creating and using technical artifacts situated in dense environments of words and things, each one with its own agenda. This way we are able to follow the contradictory forces at work in the realm of technology.

Another important source for understanding the ambiguities of today's design and use (and all the grey between the two) is the work of Michel de Certeau, whose distinction between strategies and tactics allows for the conceptualization of the ongoing power struggles on a micro level, beyond the vain debates on technology as agent of domination or progress. Such a concept of power – largely inspired by Foucault – is very much necessary if we want to situate power not only in state institutions and the marketplace, but inside the practices themselves. This leads us directly to the second element of the discussion.

3.2 Design Discourse

If we recognize software design as a pluralistic and fractured practice which takes part in shaping the fabric of the world we live in, we have to rethink our stance not only as theorists, but also as designers. Terry Winograd and Fernando Flores wrote nearly twenty years ago that "we encounter the deep question of design when we recognize that in designing tools we are designing ways of being"[18]. A dialogue with the different groups implicated in designing software is necessary in order to foster awareness of the cultural dimension of their work. This is already somewhat in the making: the open source community has adopted a very clear stance on the political issues surrounding their technical efforts and the business practices of commercial actors that embed technological creation are increasingly coming under scrutiny. The field that is lagging behind is academia. There is still very little discussion between the technical departments and the humanities and the current curricula are not fit for producing neither the "culturally-aware technologist" nor

[18] Winograd, Terry / Flores, Fernando: Understanding Computers and Cognition. A New Foundation for Design. Boston: Addison-Wesley, 1986 p.xi

the "technically-aware theorist". We hope that in the future, a more fruitful exchange will not only be possible but probable.

3.3 Policies

The third area of our discussion is policy – and luckily, there is already a very lively debate going on in this area, especially around the questions of software patents and open source. One element that seems especially important to us is the political recognition of a large part of today's creators of technology, operating outside of the classical paths, as a part of civil society. Only if we understand that writing software is one possible way of participating as a citizen, the political issues can be addressed properly. The state, as the arbiter in the ongoing battle around software patents, will have to decide, whether the amorphous coder communities sprawling on the Web that put their work at the disposition of the public domain, are of special value to society and therefore need protection against the overwhelming financial capacities of the established commercial actors. The new design practices that we tried to present and theorize in this article are by no means inevitable; although the Universal Machine is a strong base for the social and cultural activities surrounding them, the free flourishing of technical creativity is a fragile thing that can be easily be reduced to the place of mere hobbyist dabbling, as in the case of other technologies.

Digital technology and logical machinery have undermined a large part of our cultural preconceptions and, as Paul Romer once said, "a crisis is a terrible thing to waste". There (still) is democratic potential in the new metamedia and we will have to decide whether we want to nurture it or not.

Conclusion

We have entitled this paper "beyond engineering", because the term "engineering" has so much come to stand for the technocratic separation between a sphere of technology and a sphere of culture, society, and politics. We believe that in the growing diversity of technological practices lays the opportunity to overcome this artificial separation that not only seems anachronistic but increasingly opposed to a democratic idea. In a high-tech society, creating objects and tools is in part a means of free expression and therefore a highly cultural gesture. Civil society should have the right to exercise this freedom under protection of the law. Such a consensus will only be

achievable if we see technology not only as technical and economical endeavor, but as yet another form of participating in the co-creation of culture and the production of meaning.