

and meaning. And Actor-Network-Theory has shown (Latour, 1999) that actions are not properties of individual agents, but of chains linking human and non-human “actants”, combining each ones “program of action” to form hybrid actors. If we understand practice as an embedding of action in time and habit, in these views, the discursive dimension of an *art de faire* cannot be severed from its non-discursive, mechanic counterpart.

When applying this view, we see that in general, and with ICT in accelerated and enlarged form, machines are responsible for always larger parts of the action trees or action chains, rendering actions intrinsically hybrid. As a consequence, our practices have become riddled with the work of machines, in many cases without us even noticing. Software – the prime interest of this chapter – now goes even deeper than “classic” technology because many of the tasks being delegated to *logical* machinery are *semantic* in nature. Among other things, algorithms now filter, structure, interpret, and visualize information in an automatic fashion, performing tasks previously reserved for humans.

From a practical standpoint, we can understand this process of hybridization along two axes: new actions and practices are becoming possible, e.g., drawing on a virtual canvas, video communication across oceans, and real-time data-mining, and existing actions and practices are done in new ways, e.g., different in form, style, speed, efficiency, difficulty, and range.

In this sense, software is responsible for extending, both quantitatively and qualitatively, the role that technology plays in the everyday practices that make up modern life. Culture and technology are intertwined at the micro-level, to the extent that even the analytical separation of the two becomes highly problematic (Latour, 1999). Is separation between a discursive and a non-discursive level still possible when computer programs analyze email, news bulletins, and scientific publications to decide which ones to bring to our attention and which ones silently to discard? When the visibility of an opinion becomes a question of algorithms,³ meaning is deeply embedded in the non-discursive: in the software itself. Technology is not only surrounded by discourse, it is discourse. Although we do not share Heidegger’s hostile stance toward technology, his understanding of the tool as an ontological agent, as a way of “Entbergen” (revealing), is still worth considering. In “Gestell” (enframing), 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 would like to argue that technology affords not one but multiple ways of revealing being, and that the way we create technical artifacts – and software most importantly – heavily influences the cultural role they will play. Tools are not neutral; they integrate and propagate human values (Friedman, 1997). But these

³The Slashdot communication platform (<http://www.slashdot.org>) for example uses an elaborate discussion system that includes a technological measure of symbolic capital and modulates the visibility of individual messages accordingly.

values are not necessarily those of technocratic reasoning as Heidegger would have it, the whole gamut of human apprehension is possible. Software brings technology closer to us than ever before and it is time to look at the practices that spawn what has become an important part of the constitutional fabric of our cultures.

3 Software, Design and Open Source

Since the advent of modern computing in the late forties and especially the marketing of the consumer PC in the eighties, computers have come to be ubiquitous. But while the terms “computer” and “technology” have almost become synonymous and the basic technical principles have remained 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, the *Universal Machine*, coupled with peripherals like input/output devices, networks, etc., is the necessary mechanical base layer, the “specific” machine – a series of functions and procedures that manipulate information and, with proper connection, matter and energy – is the result of programming. Alan Turing stated that,

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. (1984, 4)

These words mark the technical novelty and yet another reason for the cultural significance of IT: somebody who buys a computer today gets not only the physical apparatus, but also gains access to a seemingly infinite world of logical machinery. These software programs spring from a burgeoning environment where work styles nowadays go well beyond the classical methods of engineering or even beyond the “office work” mentioned by Turing. Before we can get a closer look at these practices, we must first review some of the qualities of software.

3.1 *Properties of Software*

While there has been a continuous reflection of what software actually is, this problem is still far from being completely understood. Despite the stability of the mathematical foundations of software since Turing, Church, and Shannon, the final jury on what we can really do with it is still out. As society changes, software changes and every day there are new applications that surface around the globe. It is possible, however, to specify some of the basic properties of *logical machinery*.

Unlike other technological objects, software is immaterial. It is similar to language with respect to structure and similar to technology with respect to effect. Written as