

KITTLER IS A LIAR!

–INTERVIEWING PAUL FEIGELFELD

Paul Feigelfeld is the academic coordinator of the *Digital Cultures Research Lab* in the *Center for Digital Cultures* at Leuphana University Lüneburg. His research focuses on a transhistorical analysis of knowledge exchange between Europe and Asia. He studied cultural studies and computer science at Humboldt University, where he worked for Friedrich Kittler from 2004 to 2011. In this capacity he was directly involved with Kittler's programming projects, and he numbers among the editors of Kittler's collected works. In this discussion, Paul Feigelfeld provides insight into Kittler's practice of programming and explains how this work influenced his writings on theory and cultural studies.

Arndt Niebisch: Why did Kittler claim there is no such thing as software?

Paul Feigelfeld: Kittler is a liar. Kittler is a liar like Odysseus. Of course there is such a thing as software. Kittler himself wrote quite a bit of it. In his estate there are programs like a ray tracer, fractal sets, or radiosity software, which produces images by simulating reflections of light.

When Kittler wrote his text *There Is No Software* at the beginning of the nineties, he knew all too well what software was and also how it was programmed. In this text, his point wasn't to dispute the existence of computer code, but rather to emphasize that there must always be a materiality behind this code. With this polemic Kittler wanted to show that the hardware that so crucially defines the materiality of the computer looks like software. Which is to say that with every act of writing that software executes, the hardware specifies how this operation must look. Nietzsche's dictum, resurrected by Kittler, that our writing instruments shape our thoughts applies not only to cultural and media scholars, but also in essence to the relationship between hardware and software. I believe that Kittler's polemical position, which so strongly underscores the material foundations of technology, comes from a time in which everyone was talking about the immateriality of communication. That was a time when the Internet was coming into its own and Kittler, who had insisted so strongly on the factual, on hardware, on what actually exists, wanted to do away with an esoteric and immaterial notion of information and say: "People, don't forget the hardware! Don't forget the physis!"

Arndt Niebisch: So software does exist?

Paul Feigelfeld: Yes, but Kittler's interest in programming developed from his work with electronics. So his media practice doesn't begin with the programmable computer, but with wired and soldered circuits, where you're directly confronted by the materiality of the machine. The big project at that time was an analog synthesizer that Kittler had built together with his brother Wolf in the early eighties. A modular synthesizer, designed like the Mini-Moog, which at that time served as the basis for a hands-on engagement with technical media. It was the time of *Discourse Networks*. It was then that Kittler began drawing circuits and taking his first steps in assembly programming – he needed it for the microcontrollers he sometimes used for this synthesizer. The construction of the synthesizer still took place on paper, occasionally on typescript pages from *Discourse Networks*, which of course makes for a very nice artifact of media history. After that, both Kittler brothers bought their

first personal computer, and Kittler, rather than simply just using it as a typewriter, began concerning himself with the limits of this medium. By that I mean that the first steps Kittler took in programming weren't application-oriented; the point was rather to find out what this medium could do: how does it react when I interact with it? How can I optimize processes and squeeze the last few Hertz out of the processor? So Kittler is coming from a very hardware-related kind of programming that was only a few steps away from soldering or from manipulating technical building blocks. Writing (which is to say, the program's code) was necessary after a certain point in order to get down to the microtemporal levels of processors – where you can't play around with your fingers any more, where you need code.

Arndt Niebisch: What was the relationship between Kittler's media theory and his media practice?

Paul Feigelfeld: The role that programming plays in Kittler's media theory is in no way somehow a secondary one. In Kittler, it's that he was able to emphasize the mediatechnical *a priori* he always preached only because he himself worked in this technical *a priori*, meaning he experimented with technology at the most fundamental level possible. Kittler began programming in the eighties, and as it emerges from his own statements (but as everyone knows who studied with him, too), Kittler in fact let praxis precede theory.

Or rather, from a certain point in time when the connection between theory and practice had been established, there wasn't really a difference anymore between a discursive analysis and the attempt to synthesize a problem on the computer.

Arndt Niebisch: Kittler wasn't just a technician though, but came from literary studies. He dealt not only with technical but also with aesthetic phenomena. What is the meaning of aesthetics in Kittler's technical work or, put differently, does he have an aesthetics or poetics of code?

Paul Feigelfeld: Aesthetics and poetics play a very large role in Friedrich Kittler's programming oeuvre, especially since his programming endeavors were always about 'aisthesis' in the Greek sense, which is to say about perception. You can see that beautifully in the fact that Kittler first converted the entire theory and history of optics underpinning his book *Optical Media* into code. All the mathematics behind his graphics programs is nothing more than a theory of visual lines that later turned into a book of media history.

Poetics and aesthetics, however, also exist in Kittler's programming oeuvre on the level of language, of course, or of syntax and style. There is a style to programming, there is a particular form and a particular desire for an elegance to programming languages that also existed in Kittler, especially since he worked idiosyncratically and of course coded over many decades. Kittler didn't do this as a software developer, however, but as a humanist; he had a sometimes very funny, sometimes outmoded, sometimes very poetic manner of dealing with code that was not primarily user-oriented but which tried to get to the bottom of and fully exploit the functionality, logic, and limits of the technical medium.

Arndt Niebisch: What influence does and did this technology-related form of cultural studies have? What is Kittler's legacy?

Paul Feigelfeld: Kittler's legacy is, kind of like his book *Dracula's Legacy*, something you might call a Kittler effect. That means, plain and simple, that Kittler, in the way in which he shaped theory and entangled theory and praxis, was able to have a retro-

active effect on the disciplines from which he had initially taken so much. On the one hand, it's possible today in literary studies, philosophy, or history to think with media and also to have a kind of transdisciplinary approach oriented around the technical and not hermeneutic structures of cultural artifacts. On the other hand, however, new disciplines have evolved, too, like the digital humanities, or a new form of media studies that exists in Germany but also worldwide, and that is certainly also due to Kittler. But the important question is now how these paradigms will continue to develop. The question isn't so much "What is Kittler's legacy" as it is "What comes after Kittler"?

Arndt Niebisch: Kittler's computer code does not appear in his published work. To what extent is this part of his estate being processed and integrated into the collected writings?

Paul Feigelfeld: What's new in the collected writings of Friedrich Kittler is that we will treat source code as a kind of text on par with the other texts. That has never been done before in this way. It is only logical, though, with someone like Kittler for whom programming played such a large role. But this method also of course opens up uncharted editorial territory. Printing source code and publishing it in book form will certainly not do justice to this kind of text, and we have to consider new forms of publication in which we can be certain that the significance of programming in Kittler's oeuvre, but also the relevance of programming in the humanities in general, is portrayed clearly.

The idea, for one thing, is that there will be a book containing introductory texts by Peter Berz and me along with a programming manual Kittler wrote himself and continually cultivated over the years – a text that is really fun to read and that besides assembler code also contains quotes from Faust. On the other hand, a website is going to be created with other institutions like the Literary Archive in Marbach, where you can look at the source code and which will also have documentary materials like screenshots, for example, that illustrate how these programs function. In addition, there's a plan to develop a Kittler Linux with which you can transform your own computer for a while, as a virtual machine, into Kittler's, without the need for all too much technical background knowledge or without having to fear that you'll destroy your own computer – something, incidentally, that Kittler himself liked to do. If you use this Kittler Linux – working title: "Tetraktux," after the pre-Socratic tetractys, or tetrad, which played a big role in Kittler's oeuvre – you can sit in front of Kittler's desktop and work with his tools on his programs and other texts. As a further step, there'll be the opportunity – because Kittler was always such a staunch proponent of open source, too – to examine the entire source code in a so-called Git repository, download it, and do whatever you want with it. With this we're hoping to set in motion a certain kind of crowdsourcing with the estate, where the point is not only that a group of highly specialized scholars works on it, but rather that a democratic approach to knowledge develops.

(translated from the German by Daniel Bowles)

paul.feigelfeld@leuphana.de

Recommended citation style:

Feigelfeld, Paul. „Kittler is a Liar!“ Personal interview 13. 12. 2013. *Metaphora. Journal for Literary Theory and Media*. EV 1: Was waren Aufschreibesysteme? Ed. Arndt Niebisch and Martina Süess. 2015. Web. [Date you accessed the article].
<<http://metaphora.univie.ac.at/volume1-feigelfeld-eng.pdf>>