

FREE SOFTWARE AS COLLABORATIVE TEXT

FLORIAN CRAMER

WHAT IS FREE SOFTWARE?

Why discuss Free Software in the context of net arts and net cultures? Since about two years, Free Software—or “Open Source”—has drawn increasing attention from artistic net cultures. The *Wizards of OS* conference, first held in Berlin in 1999, was the most prolific event to bridge the gap between the arts, humanities and social sciences on the one hand and Free Software culture on the other. The politics of copyleft and free distribution of code and knowledge soon turned out to be a common ground of discourse. In this paper, I will take a different aspect into consideration by reading Free Software as a net culture and its code as a multi-layered, collaborative text. Seen as a literary practice, Free Software development is an avant-garde of writing in digital networks, and even more: Since Free Software is at the heart of the technical infrastructure of the Internet, it has—to a large extent—written its own digital network.

Definition of Free Software. In this paper, “Free Software” does not refer to “Freeware”, “Shareware” or other proprietary software given away at no cost—like Microsoft Internet Explorer, QuickTime and Real Player—, but is understood in accordance with the definitions of Free Software Foundation <http://www.fsf.org> as software which is “free as free speech, not as free beer”. Among the best-known examples of Free Software are the Linux kernel, the GNU tools and the Apache web server.

Since 1998, the term “Free Software” competes with “Open Source”, a term launched by a group around the writer and programmer Eric S. Raymond. According to this group, “Open Source” is only a different name for the same thing to gain more mainstream acceptance in the world of computing.¹ The *Open Source Definition* [Opeb] therefore draws upon the older *Free Software Guidelines* [Deb] of Debian, a non-commercial GNU/Linux distribution made by volunteers.² The guidelines can be summarized as follows:

1. Free Software may be freely copied.

Date: September 15, 2000.

This paper was presented at the conference *Interface 5* on the panel *Minor Media Operations*, Hamburg, Warburg-Haus, September 15, 2000.

¹To quote from Raymond’s *Frequently Asked Questions about Open Source*: “The Open Source Initiative is a marketing program for free software. It’s a pitch for free software on solid pragmatic grounds rather than ideological tub-thumping. The winning substance has not changed, the losing attitude and symbolism have.” [Opea]

²Both the *Debian Free Software Guidelines* and the *Open Source Definition* were originally drafted by Bruce Perens, a Free Software developer and editor of the website [technocrat.net](http://www.technocrat.net).

2. Not only the executable binary code, but also the program source code are freely available.
3. The source code may be modified and used for other programs by anyone.
4. There are no restrictions on the use of Free Software. Even if Free Software is used for commercial purposes, no license fees have to be paid.
5. There are no restrictions on the distribution of Free Software. Free Software may be sold for money even without paying the programmers.

Since the same criteria apply to “Open Source”, the two concepts indeed do not differ in technical terms. Yet each of both terms has its ambiguities: While “Free Software” tends to get confused with Freeware and Shareware,³ “Open Source” is easy to be mixed up with “open standards”—like the HTML format and the http protocol—and with software like Sun’s *Java* whose source code is publicly available, but only under a restrictive license. It is particularly important to differentiate “Open Source” and “Free Software” from open standards. While open standards are mandatory technical specifications set up by committees like the *Internet Engineering Taskforce* (IETF) and the *World Wide Web Consortium* (W3C), “Open Source” or “Free Software” developers code whatever they like for their own fun, and they are free to split their projects and develop the code into separate directions if a consensus can no longer be reached.⁴

Since misconceptions of “Open Source” are so common, I will stick with the less popular, but somewhat clearer term “Free Software”.

Free Software History. It is not accidental that history of Free Software runs parallel to the history of the Internet. The Internet is built on Unix networking technology to a large extent. Academic institutions could get Unix for a “nominal fee” including its source code in the early 1970s, and it remains to be the historical base or model of the common Free Software operating systems BSD and GNU/Linux.

The affinity of the Internet and Unix technology still persists on various level: E-Mail is nothing but the Unix *mail* command. An E-Mail address of the form *xy@z.com* is made up of what’s historically a user name on a multiuser operating system and, following the “@”, the system’s host name. This host name is resolved via the free Unix software *bind* according to the Internet domain name system (DNS); DNS itself is nothing but a networked extension of the Unix system file */etc/hosts*. Since the Internet has marginalized or even replaced proprietary computer networks like IBM’s EARN/Bitnet, Compuserve, the German Btx and the French Minitel, Unix networking technology is standard on all computing platforms.

In the 1970s, multiuser operating systems particularly attracted student hacker communities at the MIT and at the University of California at Berkeley. The

³i.e. binary-only software which can be downloaded freely and used without licenses fees (Freeware) or by paying comparatively small licenses fees (Shareware).

⁴A prominent example is the XEmacs <http://www.xemacs.org/text> editor which “forked” its codebase from GNU Emacs <http://www.gnu.org/software/emacs/emacs.htm>. The same would be impossible in open standards development. The social dynamics and institutional control of open standards development is excellently described in Jeanette Hofmanns (German) essay *Der Erfolg offener Standards und seine Nebenwirkungen* [Hof99].

concepts of open, decentralized computer networks and free multiuser operating systems have their origin in the computer science labs of these institutions. While the MIT hackers wrote their own operating system ITS and the Berkeley hackers improved and extended the original Unix codebase, their “hacks” eventually evolved into:

1. the BSD family of operating systems with the free versions FreeBSD, NetBSD and OpenBSD. All of them use a codebase that was originally developed in Berkeley under the project leadership of Bill Joy.
2. the GNU/Linux operating system. All major Linux-based operating system distributions—RedHat Linux, SuSE Linux, Turbo Linux, Debian GNU/Linux, Mandrake Linux, Corel Linux OS and Caldera OpenLinux, to name only a few—build on the GNU software written since 1984 by the Free Software Foundation (FSF) and on the Linux kernel written since 1991 under the project leadership of Linus Torvalds.⁵ The FSF was founded and is still being led by former MIT hacker Richard M. Stallman.

Open technology has been a key factor for the acceptance of computers and networking: The open architecture of the IBM Personal Computer made computers cheap and popular since the 1980s, and with the open architecture of the Internet, global networking became popular in the early 1990s. Lately, Free Software has made high-end Unix server computing available to anyone willing to learn the technical details. Whether Free Software can become as popular on mainstream desktop computers and eventually de-commoditize all computer software, remains to be seen, but is not the question I want to investigate here.

FREE SOFTWARE AS A NET CULTURE

In the middle of the 1990s, “net culture” became the keyword for artistic, art-critical and political discourse in the Internet. The term was closely identified with mailing lists like Nettime <http://www.nettime.org> and Rhizome <http://www.rhizome.org>, conferences like the one where I present this paper and print publications like the Nettime anthology [BMBB⁺99]. “Net culture” used to be pronounced as a singular noun in these forums and media referring only to the discourse they created.

Free Software is an outstanding example that there is not one, but many net cultures. It predates artistic net cultures in the Internet by roughly twenty years. The Free Software copyleft can be seen as the quintessential reflection of this long experience. Invented to preserve the traditional academic-artistic freedom of speech and citation in the digital realm, the copyleft has radically rewritten it nevertheless. The concept that code, i.e. text, may not only be freely copied, but even modified (“patched”), willfully recycled and commercially redistributed

⁵There is an ongoing debate in Free Software culture whether operating systems based on the Linux kernel should be called “Linux” or rather “GNU/Linux”. In order to be functional at all, a “Linux” setup relies upon the GNU C Compiler (gcc) to translate all program sourcecode into machine-executable binary software, the GNU C Library (glibc) as the interface between the Linux kernel and userspace applications, and the GNU tools for the basic user commands. Although it is possible to replace at least the GNU tools and the glibc with non-GNU workalikes, all common “Linux” distributions use the Linux + GNU software setup. I will therefore stick with the name “GNU/Linux” where I refer not only to the kernel, but to the whole operating system.

by anyone without the author's permit is foreign to the post-medieval Western arts and sciences. In print culture, such practices are considered plagiarism and theft.

Even for the digital net arts, the copyleft remains an unresolved challenge. Many, if not most net artworks depend on proprietary authoring and display software,⁶ and the distribution terms of their code are rarely clarified.⁷ Yet Free Software has as subtly as significantly influenced the digitally networked arts. Without free E-mail server software like Majordomo <http://www.greatcircle.com/majordomo/> and Sendmail <http://www.sendmail.org>—and the overall possibility to set up inexpensive servers using the GNU/Linux and BSD operating systems on stock PC hardware—, the artistic net cultures of *Nettime* et.al. hardly could have operated non-commercially and with free participation.⁸ Friedrich Kittler's observation that artistic tools conceptually shape what is made with them [Kit85] also applies to the net arts. The fact that *Majordomo* and *Sendmail* became major tools of artistic net activity is an important—but of course not the sole—explanation why contemporary *Net.art* tends towards conceptual, discursive and text-heavy work instead of the immersive “virtual reality” environments many critics had expected them to deliver. The latter would have required expensive proprietary software for design and display, closed high-speed networks and, as a result, dependence on highly funded institutional infrastructures, limited community participation and top-down instead of bottom-up organization of this particular net culture.

FREE SOFTWARE AS WRITING

The relevance of Free Software for other net cultures is not limited to the tools it has created and the infrastructures it has made possible, simply because those tools themselves are the very object of Free Software culture: they are *text*, results of complex textual processing. Moreover, this text is being produced with tools which themselves are free code.

While the phenomenon that text is being built with tools which are source text themselves applies to the proprietary software as well, there is an important difference: Free Software source text is not withdrawn from the public. It cannot be abandoned by company management and does not disappear when development has ceased. All Free Software builds up to a public repository of text-coded, free-to-use knowledge. It accumulates to an archive. Instead of being written from scratch, new Free Software can be built from whatsoever is in that archive. Free Software therefore is highly intertextual. Free Software development is the earliest and still most successful practice of collaborative writing in computer

⁶Such as Macromedia's *Shockwave* and *Flash* in “Net.art”, Opcode's *MAX* in electronic music and Eastgate's *Storyspace* in hypertext fictions.

⁷The artist group 0100101110101101.ORG <http://www.0100101110101101.org> put this issue up front when it mirrored and partially modified well-known *Net.art* web sites on its own web site.

⁸Early artistic computer networks like the Thing BBS <http://www.thing.net> charged their subscribers (at least in Berlin) before they migrated into the Internet.

networks. With its system of textual production and politics of code, Free Software is by far the more advanced net literature than what is commonly understood as net poetry and net fiction.⁹ Free Software may be seen simultaneously as

- a freely accessible, ever-growing body of code—a text archive;
- recursive (i.e. self-applied) text processing, since available text is used both as a source *and* as a building tool to create new code;
- text processing even through the *medium* of text, because Free Software development infrastructures mostly depend on mailing lists and command-based version control systems.
- a “hacker” culture which advocates freedom of information and codes its politics into the legal texts of the copyleft.

The coded copyleft might be the clearest interstice between Free Software as a net culture and Free Software as net text. Both these aspects already come into play when Free Software is being written. Free Software development is typically achieved by self-organized volunteer projects whose members communicate and collaborate via the Internet. The development work consists of:

1. Writing program source text

This involves evaluating of available Free Software source code for possible inclusion and adaption. It also involves picking—and compiling—the coding tools which themselves are Free Software source text.

To accommodate its own needs, Free Software has developed the arguably most sophisticated writing tools for the distributed authoring of text. Particularly outstanding is the *Concurrent Versioning System* (CVS) [Ced99] which allows authors to take portions of text—regardless whether it is written in programming language or in natural language—over the Internet, work on them at home, and synchronize the changes with the revisions of other collaborators any time. CVS-based writing might be the technically most radical departure from the typewriter-and-mail paradigm in text editing to date.

2. Writing documentation text

Documentation is both internal and external to the program source text when the latter contains annotations and separate reference documentation is being written.

Free manuals remain a political issue within Free Software development. A number of companies base their business model on giving away the software under free licenses and charging for documentation and support.¹⁰ In the ideal case however, a second textual recursion occurs within in Free Software which is common in all modern knowledge systems since Diderot’s and d’Alembert’s *Encyclopédie*:¹¹ The text teaches the reader all

⁹How net literature—“hyperfiction” and “new media poetry”—relates to poetic practices rooted in programmer’s cultures is discussed in more detail in my (German) paper [Cra00].

¹⁰Among those companies are *O’Reilly* publishers, *Sendmail Inc.*, *VA Linux*, *Scriptics*, *Helix Code* and *Eazel*. All of them are involved in the development or documentation of critical components of GNU/Linux operating systems.

¹¹I thank Wau Holland for pointing this out to me in a preparatory meeting for the first *Wizards of OS* conference.

steps which were necessary for its creation so that all the information it contains may be re-applied to itself.

3. Communication over mailing lists, bugtracking systems and IRC

Free Software development teams almost exclusively constitute themselves and communicate over the Internet, in mailing lists and on IRC servers. Interpersonal communication therefore is a third layer of text which regulates the design of both program and documentation source text. It operates as a cybernetic feedback loop for the development process.

4. Writing legal text

Free Software is legally defined. It is software under certain licenses, i.e. legal documents. The most common types of copyleft include the GNU General Public License <http://www.gnu.org/copyleft/gpl.html>, the BSD License and the Perl Artistic License. Whether program source text is free solely depends on whether it is copylefted. Legal text therefore is the fourth layer of text regulating the entire flow of text generated in Free Software projects.

Free Software is thus a highly sophisticated system of recursive text generation for a public pool of knowledge. It is text code created from text code with text-coded tools and textual communication over networks. The types of texts processed in Free Software are extremely diverse: They include executable binaries,¹² text written in programming languages, text written in natural languages for documentation, text written in natural languages for communicating and steering development, and legal texts defining the fair-play rules of the recursive textual processing.

OBJECTIONS

Both the Free Software engineering and the net artistic camps are traditionally skeptical about attempts to read Free Software in terms of the net arts. The objections were particularly voiced when the Linux kernel was awarded the Golden Nica in the “net” category of *Ars Electronica* 1999. At the *Wizards of OS* conference in the same year, the net artist Alexej Shulgin argued that Free Software is “functional” while *Net.art* is “non-functional”, self-sufficient code.¹³

I do not find this point viable from an analytical perspective, since the division between “functional” and “non-functional” is purely arbitrary and subjective. I/O/D’s *Web Stalker* [I/O97], an experimental Web browser and well-known *Net.art* work, is arguably more “functional” than the teddy bear desktop emblem *xteddy* which is contained in all major GNU/Linux distributions. Moreover, the distinction between “functional” Free Software and “non-functional” *Net.art* falls back into late-romanticist notions of the absolute artwork versus

¹²Which can be read as “text” if text is linguistically and semiotically defined as a finite number of discrete signs chosen from a finite set of signs. In computing, “text” is rather colloquially understood as code from natural-language alphabets as opposed to binary code. Being a philologist, I refer to the prior concept of “text”.

¹³According to [Bos98], the label *Net.art* was coined in 1996 by the net artist Vuk Cosic and has been associated with a particular generation of net artists since (involving, among others, Cosic himself, Heath Bunting, Olia Lialina, Alexej Shulgin, jodi and I/O/D).

lower craftsmanship. It also neglects that with its multiple self-applications of text, the development and use of Free Software is to a large extent its own purpose. No other operating system is as open and seductive to be used as an end to itself as GNU/Linux.

Just as arbitrary as the distinction between “functional” and “non-functional” software is that between program source code and poetry. To date, all attempts to formally define poetry and poetic language have failed. The decision whether a text is poetry will always be up to the reader. The notion of “program code” versus “poetry” was first put into question by the French poet and mathematician François le Lionnais, who co-founded the Oulipo group with Raymond Queneau. In 1973, le Lionnais released a volume of poetry written in the programming language Algol. The practice has been revived in the 1990s by people who write poems in the Perl scripting language.

CONCLUSION

Read as a net literature and a net culture, Free Software is a highly sophisticated system of self-applied text and social interactions. No other net culture has invented its computer code as thoroughly, and no other net culture has acquired a similar awareness of the culture and politics of the digital text.

Much *Net.art*, net literature and critical discourse about them has focused on the aesthetics and politics of desktop user interfaces. In its focus on code, Free Software shows that net cultures are about more than just what is between people and the network. To date, it remains a rare example of electronic literature which does not confuse the Internet with web browsers.

(Acknowledgement: This paper was written using the Free Software programs L^AT_EX, E^T_X, bibtex, bibtools, pdflatex, latex2html, lynx, XEmacs and GNU Ghostscript on an office and a home PC running Debian GNU/Linux with reiserfs, XFree86 and larswm. Thanks to Ronda Hauben for some corrections of the section on Free Software history.)

REFERENCES

- [BMBB⁺99] Josephine Bosma, Pauline van Mourik Broekman, Ted Byfield, Matthew Fuller, Geert Lovink, Diana McCarthy, Pit Schultz, Felix Stalder, McKenzie Wark, and Faith Wilding, editors. *Readme! Filtered by Nettime*. Autonomedia, Brooklyn, 1999. 3
- [Bos98] Josephine Bosma. Is It a Commercial? Nooo... Is It Spam? ... Nooo - It's Net Art. *Mute*, 10:73–74, 1998. 6
- [Ced99] Per Cederqvist. *Version Management with CVS*. Signum Support AB, Linköping, 1992-1999. <http://www.lorai.fr/~molli/cvs-index.html>. 5
- [Cra00] Florian Cramer. Warum es zuwenig interessante Netzdichtung gibt: Neun Thesen, 2000. http://userpage.fu-berlin.de/~cantsin/aufsaetze/netzliteratur/karlsruher_thesen.pdf. 5
- [Deb] Debian Project. The Debian Free Software Guidelines. http://www.debian.org/social_contract. 1
- [Hof99] Jeanette Hofmann. Der Erfolg offener Standards und seine Nebenwirkungen. *Telepolis*, 7 1999. <http://www.heise.de/tp/deutsch/special/wos/6453/1.html>. 2
- [I/O97] I/O/D. I/O/D 4: The Web Stalker, 1997. <http://bak.spc.org/iod/>. 6

- [Kit85] Friedrich Kittler. *Aufschreibesysteme 1800 1900*. Fink, München, 1985. 4
- [Opea] The Open Source Initiative. Frequently asked questions about open source. <http://www.opensource.org/faq.html>. 1
- [Opeb] The Open Source Initiative. Open source definition. <http://www.opensource.org/osd.html>. 1

C/O FREIE UNIVERSITÄT BERLIN, SEMINAR FÜR ALLGEMEINE UND VERGLEICHENDE LITERATURWISSENSCHAFT, HÜTTENWEG 9, 14195 BERLIN

E-mail address: cantsin@zedat.fu-berlin.de