

The Politics of Magic Pixie Dust.

Socio-technical Effects of Open-Sourcing Netscape Communicator Code.

By Alek Tarkowski (atarkows@ceu.edu.pl), Central European University, Warsaw, Poland.

*You can't take a dying project, sprinkle it with the magic pixie dust of "open source",
and have everything magically work out.*

Jamie Zawinski, Netscape developer

On January 22nd, 1998, Wired News reported that "This morning, Mountain View headquarters announced it would freely license not only the current versions of Netscape Navigator and Communicator, but also make the source code for the products freely available on the Net" (Oakes 1998f). Two reasons were mentioned: the company "wants to be perceived as friendly, open, and believing that code should be free" (Oakes 1998f) and is "angling to increase its market penetration and mind share" (Court 1998a). This paper is a look into the politics of freeing or open sourcing software code. The case of Netscape was unusual, since for the first time a commercial software was being open sourced. Application of the Actor Network Theory (ANT) framework to the developments of a socio-technical system with a software core allows both to understand the transformation of software from proprietary to F/OSS¹ and to show specificity of software as a materiality, in ANT categories. I argue that the Netscape case shows that open sourcing can be a complex process that cannot be summed by the slogan "information wants to be free". This example allows us to critically examine the concept of the "F/OSS community", as well as to consider in what manner F/OSS programming can be considered a social movement.

A short history of Netscape

Netscape Communications Company is a WWW browser developer, which currently defines itself as "aiming to be the leading provider of open software that links people and information over the Internet and intranets"². The company was created in mid 1994 by Jim Clark, creator of Silicon Graphics Inc., and Marc

1 As will be later explained, significance of naming conventions is very high in the case of free or open source software. To remain neutral to these political choices, in this paper we will use the acronym F/OSS (Free / Open Source Software).

2 <http://wp.netscape.com/company/index.html>

Andreessen, a student. In 1993 the later, with a team of programmers at the U.S. National Center for Supercomputing Applications (NCSA) created Mosaic, a freely available graphical user interface that provided “a consistent and easy-to-use hypermedia-based interface into a wide variety of information sources” (Andreessen 1993) and thus opened the Internet to the masses. Netscape managed to hire key Mosaic programmers and decided to create an improved proprietary version of the browser. Officially, the software was named Netscape, but internally the name Mozilla, made up by one of the developers, Jamie Zawinski, was used for what was seen as the “Mosaic Killer” application. The name was also given to the company mascot, a green Godzilla-like lizard whose image dominated company’s early web pages. At the time of the first release, Andreessen commented that “Making Netscape freely available to Internet users is Netscape Communications’ way of contributing to the explosive growth of innovative information applications on global networks” (Mendelson). The company was giving to the community, but also aggressively marketing its own product. Through widespread free distribution, Netscape managed to make the browser a proprietary standard. At the end of the year Netscape controlled 80 percent of the browser market and a non-free version of the browser was released, generating a revenue stream for the company. In the meantime, Mozilla mascot was replaced with a different, starry logo. In August 1995, Microsoft released Internet Explorer. 1996-1998 were the years of so-called browser wars, when both companies competed for the market and Netscape gradually lost its dominant position to Explorer, which was free and came bundled with the Windows operating system. Andreessen believed that a way to stay ahead of the opposition is to release a new version of the software every six months even if it is a *beta*, an unfinished product.

By 1998, it became clear that not only Netscape browser will not become a software standard more important than the PC operating system, as was hoped for in 1994, but will lose the so called “browser wars” too. In one year, Netscape’s share fell from 70 to 59 percent and Internet Explorer controlled in 1998 already 37 percent of the market (Court 1998a). Jamie Zawinski, one of the key Netscape developers, later wrote that “Microsoft had succeeded in destroying that market. It was no longer possible for anyone to sell web browsers for money. Our first product, our flagship product, was heading quickly toward irrelevance.” Netscape has hit “one of its blackest periods” (Zawinski 1999) and lay-offs of as much as 1/8th of 3200 employees begun as part of a restructuring effort. At the beginning of the year,

Netscape presented its new, two-part strategy. One change was conservative, a shift of focus from individual to corporate market in an attempt to gain revenue from server software sales and advertising revenue from the Netcenter homepage, which was redesigned to serve as “a service-based gateway into the Web for corporate customers”(Wired News Report 1998c). The second was radical: Netscape would not only once more distribute its browser for free, but furthermore make browser’s source code available for others to freely modify and distribute. Wired called this “taking the idea of a give-away browser even one step further than Microsoft” (Court 1998a).

Thus Netscape decided to join the F/OSS movement. For the first time since the distinction between proprietary and free software was made, major commercial code was being transferred to a F/OSS distribution license, providing its users with new technical freedoms. The company could only predict that software, which by itself did not provide revenue, could become a source of income in an alternative revenue model.

Free and Open Source Software: a note about terminology.

The term “open source”, similarly to the term “Internet”, is often used in an imprecise manner. In both cases we are dealing with a socio-technical imbroglio (Latour 1998), a network of heterogeneous elements weaved around a technical core, one in which both humans and non-humans are actors, “entities that do things” (Latour, quoted in Stalder 1997). In common parlance, the most important distinction between the whole network of the F/OSS movement and its technical core, the software itself, is often blurred. The term F/OSS can be used to describe software with free access to source code, providing a range of freedoms for the user; a model of collective work, in which access to the code allows increase in creativity and efficiency; a business model, on which a F/OSS segment of the programming market has been built; and a social movement that considers programming a political activity and ties social values to F/OSS code.

This is a symptom of the fact that the relation between development and politics of computer code and more broadly understood social and political processes taking place in technology-saturated societies are not well understood. An outdated theoretical model is, according to Bruno Latour, one of crucial obstacles to this goal: “social scientists are always a war late, seeing devious political plots behind techniques, when the socio-techniques allow [...] to add new fresh unexpected ways of redefining power” (Latour 1998). Latour blames for this primarily the duplicity of

analysis, which divides socio-technical systems into two separate homogeneous spheres of the social and the technological, and often reduces the later to the former.

Against common use, choice of wording, mainly between terms “free software” and “open source software” becomes a political decision inside the programming community. “Free software” is a term introduced in the eighties by Richard Stallman, founder of the Free Software Foundation and father of the GNU project. Stallman stresses that “<<Free software>> is a matter of liberty, not price” (FSF 1996). According to the definition, free software provides the user with four freedoms: the freedom to run the program, for any purpose; the freedom to study how the program works, and adapt it to your needs; the freedom to redistribute copies so you can help your neighbor; and the freedom to improve the program, and release your improvements to the public, so that the whole community benefits”(FSF 1996). Among consequences of free software are greater user freedom, increased creativity and rebirth of a community of programmers united by common work. Free software was not only a programming strategy, but a moral and social project.

The term “open source” has been devised in 1998 by a group of programmers from the newly formed Open Source Initiative (OSI), who supported a less adversarial stance and believed that the new, pragmatic strategy requires a new brand name, new label. To meet the definition of open source software, a distribution license must: allow free distribution; include source code with the program and allow its distribution; allow modifications and derived works, under same terms as the original license; and not discriminate against persons, groups, or fields of endeavour.

Free software and open source are two labels for the same sort of software, available not only as an executable program, a “black box” of Assembler code called a binary, which is unintelligible for humans, but also as source code that developers can read, modify and compile into a binary. But these labels represent two widely different views on politics of code. According to Stallman “Open source is a development methodology; free software is a social movement”(Stallman 1998). For open source supporters proprietary software is solely a non-optimal solution. For the free software movement it is, as Stallman puts it, a social problem.

A Short History of F/OSS

In a period directly after World War II, when development of computer and network technologies started, a techno-elite of scientists and researchers created

software that was free and freely available. This was partially due to the fact that research projects were funded by large, monolithic institutions that did not care about commercial gains, and partially stemmed from an institutional culture grown from the tradition of scientific research. For the early computer engineers development of science and knowledge was the highest value, while collective work and exchange of knowledge were a condition of increased creativity that could serve this value. There were also pragmatic reasons, as free access to source code ensured compatibility of versions ported to different systems. IBM computer giant initially focused on hardware sales, did not consider software a revenue source and assumed that improved, due to availability of source code, programs, can increase hardware sales. Importantly, the operating system Unix, around which in the eighties developed what Eric Raymond calls a “network nation” of hackers, was freely available in the seventies, as anti-monopoly regulations prohibited AT&T, American telecom monopolist at that time, to profit from sales of the system.

Logic of free software started to crumble at the end of the sixties. Accused of unfair competition on the hardware market, IBM divided hardware and software sales, in what was the symbolic birth of the software industry. Microsoft, founded in 1975, was one of the first companies that profited solely from writing and selling software. In 1984 the AT&T monopoly broke and the company began collecting license fees for the Unix system, to which it retained rights.

Development of non-proprietary software continued in academic and non-governmental institutions. Importantly, key Internet protocols, including the TCP/IP protocol that in technical terms defines the Internet, are until now developed as open protocols by the Internet Engineering Task Force (IETF).

On the West Coast, researchers at the University of Berkeley in response rewrote the code and created an unencumbered version of the system: the 386BSD system. On the East Coast, Richard Stallman, a programmer working at the MIT AI Lab became dissatisfied with AT&T's commercialization of Unix and angry at what he saw as a decline of a community that flourished through F/OSS work. To Stallman, proprietary rights were an impediment to effective software development. In 1984, Stallman gave up his former job and began work on a F/OSS type Unix-like system called GNU (GNU is Not Unix). One year later, he founded the Free Software Foundation (FSF), an organization producing and distributing free software. . Stallman developed a range of important F/OSS software, although the GNU system lacked its key component, the kernel. It seems that Stallman's most significant

achievement was a piece of legal and not software code: the GNU GPL (GNU General Public License), a legal document that ensured the status of F/OSS. Stallman devised what he called “copyleft”. According to the license, the author retains copyrights to the code, but uses them to make the software freely available. GNU GPL is unique among other licenses in its viral property, which mandates that any modifications or other code incorporating GPL code be also released under the license – this ensures that code that begins free remains free (Behlendorf 1999). While free software ideology is often considered a radical position in the F/OSS movement, the GNU GPL license is used by a majority of projects, leading a Wired magazine commentator to comment that “Stallman may have pulled one of the greatest hacks of all time by creating not a work of software, but a legal document” (Stutz 1998).

Both GNU and 386BSD were distributed under F/OSS type licenses and together gave life to F/OSS as a legal position and distribution policy. FSF stance was especially zealous and motivated by an anti-commercial ideology. In 1991, the 386BSD distribution of the Unix system was created and became the first free software platform, protected by a F/OSS license.

At the same time, Linus Torvalds, a computer science student from Helsinki, began writing one and used the maturing Internet network to start a distributed, collaborative project with hackers³ working for free, motivated by the intellectual challenge and a reputation economy. Torvalds presented the first version in 1991 and after three years a stable version of the Linux GNU kernel was released, against existing assumptions that large-scale software development cannot be successful. Linux proved to be an important management and institutional experiment, which proved that tanks to the Internet, F/OSS can be a successful coordination and cooperation strategy that produces extremely stable and bug-free code. The Linux case provided pragmatic rather than ideological arguments for the movement.

In 1997, Eric Raymond, a renowned software developer and maintainer of the Hacker Jargon File, an online repository of developer culture, turned ethnographer. In his essay “The Cathedral and the Bazaar”, he presented an analysis of free software process as both a healthy business model and successful collective work. He created the metaphor of the cathedral and the bazaar: proprietary software, built like cathedrals, by small, closed teams, was compared with development bazaars that

³ A hacker is a slang term for a developer, meaning someone who programs enthusiastically or even obsessively and enjoys exploring details of programmable system. Term hacker can be generalised anyone who excels at his work and enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

the free software projects were. Raymond was a clear supporter of Torvalds's pragmatic stance and helped to spread it through his texts, which he describes as “a bit of memetic engineering on the hacker culture's generative myths” (Raymond 1999b).

In 1998, a group of prominent hackers, including Raymond and Torvalds, decided to reframe the F/OSS movement so that it could be accepted by the mainstream of the software sector. The Open Source Initiative (OSI) was formed to promote the term “open source” as an alternative to “free software”, which was considered ambiguous but also overly ideological - “the term makes a lot of corporate types nervous”, observed Raymond (Raymond 1998b). The change was to be the proof that developers are “willing to work with and co-opt the market for our own purposes, rather than remaining stuck in a marginal, adversarial position” (Raymond 1998b).

The 1998 open-sourcing: a look behind the screen

Transformation of Netscape Communicator into F/OSS, upon closer scrutiny, reveals itself as a complex and political process. The switch to a F/OSS model was neither simple nor instantaneous. Actors were only partially involved in the process and tried to simultaneously attain other goals.

Supporters of F/OSS argue that open software and protocols ensure existence of the Internet in its current state, as an end-to-end network with intelligence located at the ends and dumb in-between network that at the same time provide a neutral communication channel and remains autonomous from external economic or political pressures. Presence of F/OSS makes it impossible to achieve dominance of a proprietary standard controlled by a commercial entity. In the case of browser software, which remains one of the key software “building blocks” of the Internet, if Microsoft could achieve dominance in this market, it could gain control over the Internet by building proprietary channels upon open Web protocols. These channels would be serviced by Microsoft servers and accessible only through Microsoft's proprietary software. A precedent existed in the form of Apache, a F/OSS type server software. Strong position of Apache forced software vendors to solve interoperability issues by conforming to open standards. Without Apache, companies could implement incompatible, proprietary improper protocols in own servers and clients, which could ultimately cause the World Wide Web to fork.

At first, open sourcing Netscape code seems a solution to such a problem. But

firstly, this required Netscape to achieve a significant market share – a condition that largely depended on its customers. Secondly, protection of Internet's autonomy and freedoms afforded by the network was neither the sole or key goal of Netscape company.

The decision to release the source code was influenced by a couple of factors. An internal white paper by Frank Hecker, which circulated in the company, advocated this move and sketched a series of alternative business models based on F/OSS software distribution. As important was Raymond's work, which Netscape CEO Jim Barksdale described later as "fundamental inspiration for the decision" (Raymond 1999). The move also received support from the "engineering pit", where developers were accustomed to the idea of F/OSS since Java and HTML code, with which they worked, was of similar nature. Netscape also had experience with running against conventional wisdom, when in 1994 it gave out Netscape browser for free, a move "seen in retrospect as a successful innovation that was a key factor in Netscape's rapid growth" (Hamerly, Paquin and Walton 1999).

Preparations for a F/OSS distribution of the browser involved changes in the structure of both software and Netscape as an institution. The source code was made more accessible to outside developers and third party modules that remained proprietary were disposed of. Legal issues were solved through creation of two new distribution licenses which leveraged Netscape's rights as the former owner of the software with F/OSS goals and values. A decision was made to rename the software and Mozilla was chosen as the official name for the family of Netscape's F/OSS browsers. Netscape reserved the right to use the developed code in proprietary software, which would keep the brand name of Netscape Communicator. Finally, coordination issues were solved. The day Netscape announced its plans, Jamie Zawinski registered the domain mozilla.org and drafted a preliminary structure of an organization, which would fulfill the role of the code's maintainer. The maintainer coordinates development of a project and has the final word if conflict arises. In F/OSS projects, maintainer's authority rests on a complicated mix of charisma, successful development history and "evolutionary success" of maintained code (Weber 2000). Such is the case of both Linus Torvalds, original developer and now "a maintainer" of the Linux kernel as well as the group of original Apache server developers. In the case of Netscape, there was no evolutionary development: maintainer organization was created in a "top-down" manner while as part of project's institutional infrastructure. Mozilla.org became an independent non-profit

organization, staffed and provided with resources by Netscape.

Reflexivity of participating actors distinguishes the Netscape case from earlier F/OSS developments. In 1998, 386BDS and GNU were well known stories and analysis such as Raymond's "The Cathedral and the Bazaar" or Frank Hecker's "Setting Up Shop" were available. This reflexivity allowed Netscape to set up a structure that earlier emerged spontaneously – a task that would not have been possible earlier in F/OSS history. The Mozilla project replicated some of the spontaneously emerging divisions. For example, by hiring a team of Mozilla programmers, Netscape replicated the division between core programmers and a wider community, present in F/OSS communities.

Despite the differences, one thing put Netscape in line with other successful F/OSS projects was the strong foundation in the form of existing Netscape browser code. Observers commented that Netscape has taken "a developer-driven concept and implemented it in a developer-driven style: with strong support and a responsive developer policy" (Oakes 1998a). Developers saw a sign of Netscape's commitment to this policy when the company took into account their critical input while drafting the Mozilla Public License (MPL) distribution license. Missing was only a motivated developer community to which Mozilla.org and Netscape were set to cater.

Actors involved in the open-sourcing process

Upon closer scrutiny, we can recognize four key actors involved in the process of turning Netscape code into F/OSS – and none of them simply wanted to "Free the [Mozilla] Lizard".

Netscape Communications Company wanted to benefit commercially from making the code free. Far from becoming altruist, Netscape was trying to turn a seemingly non-commercial development model into a business strategy. That this is possible was argued by Hecker in his white paper (1998) and at that time there were first signs of companies making profit from the Linux operating system.. Jim Hamerly, vice president of Netscape's client products division, called this mixture of business and community goals as an example of "give and take of the Net"⁴ (Oakes 1998e). Netscape believed that this was "one of the only ways of scale to compete against Microsoft" and that "it can make that model a huge commercial success", according to company's spokesman (Oakes 1998f). According to the new business plan, client

⁴ This is a sign of its times: late nineties, when the Dotcom boom ended the period when Internet was an autonomous zone and made it a space of struggle, with friction appearing in places where smooth and striated spaces meet.

sales were only needed to boost server sales and open sourcing of code was “a creative way to achieve an important objective: increase business customers' confidence in Netscape” (Court 1998a). It must be noted that Netscape truly adopted the F/OSS rhetoric and in this it differed dramatically from Microsoft, which at that time stated that “It really hasn't arisen, the need for source code” (Oakes 1998 a).

F/OSS evangelists consulted by Netscape had their own vested interests. Raymond believed that “helping Netscape's gamble succeed had just become a very high priority for the hacker culture” (Raymond 1999). For him, Netscape cracked the barrier dividing the “hacker ghetto” from the business world. An effort had to be made to ensure that Netscape would succeed and furthermore wouldn't be considered a “unique one-off”. Raymond became what he termed an “accidental revolutionary” and soon “twenty-odd community leaders” joined him in OSI, a marketing project of sorts. Raymond wrote: “Our success after Netscape would depend on replacing the negative FSF stereotypes with positive stereotypes of our own - pragmatic tales, sweet to managers' and investors' ears, of higher reliability and lower cost and better features” (Raymond 1999). The Netscape move was applauded, but quickly afterwards software community and industry focused upon OSI, which was promoting its new “open source” brand, as well as on Linux, which was the true champion and demonstration case. “Yes, there are other things going on in the open-source world, and the campaign will bow respectfully in their direction - but Linux started with the best name recognition, the broadest software base, and the largest developer community” (Raymond 1999).

We can assume the existence of a rather stable group of hackers residing in Netscape's “engineering pit”, with a large group present since company's beginnings. They are symbolized and represented in the media by Jamie Zawinski, “20th person hired by Netscape”, an important coder in the original Netscape project, father of the Mozilla icon and later leader of the Mozilla project. In a manner typical of hackers, Zawinski combines espousal of liberty and individualism seen as supreme values, with love for innovative, efficient code. Speaking about the original browser, he stated: “We kick-started a new communications medium. We changed the world” (Zawinski 1998). For Zawinski, between 1994 and 1998 Netscape mainly grew big and because of that stopped being creative, became a “slow and stupid union” of people, “was shipping garbage, and shipping it late”. “The magic was gone” (Zawinski 1998).

Gina Neff and Gerard Stark use the name *permanently beta* to describe “a fluid

organizational form resulting from the process of negotiation [...] of design”, often present among software companies, which require constant responsiveness and adaptability (Neff and Stark 2002). Amidst the constant change, companies need to ship regularly new versions, so that market and clients see that work going on is not without effects and that their needs are met. Dynamics of the industry enforce a dizzying rate of production, an example of a broader range of phenomena that are all adaptations to the difference between human and machine speeds. Just like teenage groups, in which constant, mutual exchange of short text messages (SMS) is often a strong norm, companies have to ship often to remain in the market. “Release early, release often” was a principle adopted by Torvalds, which is pointed out as one of the sources of success of the Linux project in its early phases. Since 1994, Netscape was regularly “kicking through the door” new releases, but the *permanently beta* state transformed with time into a ritual and was no longer a source of innovation. The 1998 decision meant for Zawinski that the company is again doing something daring and motivations behind the move did not really interest him: “An act of desperation? Perhaps, but still a very interesting and unexpected one”. Zawinski represents people intimate with the Netscape code, for whom the specific piece of software was a value. “The fact that Netscape was no longer capable of building products wouldn't matter: the outsiders would show Netscape how it's done”. The change allowed him to continue, under much more favorable conditions, to continue his work: “I saw it as a chance for the code to actually prosper”. Therefore Zawinski involved himself both in grassroots evangelism inside the company and took upon himself the job of creating the Mozilla.org institution.

The fourth group is the developer community, which before the code was released represented the weakest link in the network, an unknown variable. There was no place for them in the process managed in a top-down manner by Netscape.

ANT assumes that social order is achieved through a range of patterning processes called translations. Translation implies transformation with the possible effect that one thing may stand for another (Law 1996). It is a process, in which “identity of actors, the possibility of interaction and the margins of manoeuvre are negotiated and delimited” (Callon 1999, 203). Actors, in making others dependent upon them, “translate their will into a language of its own” (Stalder 1997). A translation of power relations in a heterogeneous network requires, in turn: 1. problematization: selection of a set of actors and definition of their identities; 2. interestment: attempt to impose and stabilize actors' identities; 3. enrollment:

attribution of roles, a result of successful interestment; and 4. mobilization of allies by their displacement in space and time (Callon 1999). One has to note, that in Computer Mediated Communication the mobilization has more to do about competing for actor's limited attention than with actual displacement, thus lowering significantly costs of involvement. The translating actors, or spokesmen, attempt to define themselves as an obligatory point through which all relations must pass and at the same time define other actors as indispensable to the project. In the Netscape case, while the identities of the developer community were stable, Netscape could not be sure, whether enrollment and mobilization would be successful.

Although developer preferences were unclear, Netscape representatives seemed confident. Andreessen told "Wired" magazine that he expects to "see a tremendous amount of feature innovation" (Oakes 1998a) and Netscape's client product manager said he expects "new and better features, add-ons and related products" (Oakes 1998g). Predicting effects of collaborative browser development, "Wired" commented in a text titled "Netscape: bring on the Frankenbrowsers" that "the very concept of a browser could be morphing before our eyes" (Rein 1998) with browsers assembled on the fly according to individual preferences. Still others were skeptical whether developers will get involved and Microsoft declared that developers, with whom it collaborates, actually prefer to work with binary versions of software provided by the company. .

Steven Weber (2000) presents what he calls the signaling argument, which could explain eventual failure of the Netscape project. Peer recognition in F/OSS systems provides developers with renown, which eventually could lead to commercial employment and financial rewards. Participation is a demonstrative act showing quality of one's own work. Signaling incentive is stronger when performance is visible, when it has high impact and yields good information about one's talent. Such conditions are present in "more technical settings like operating systems and not in end-user applications" and therefore it is the former that, according to Lerner and Tirole (2000), draw more flourishing F/OSS projects.

Where is the Software?

Thomas Hughes considers every socio-technical system a "seamless web", in which "the social is indissolubly linked with technological and economic" (Law 1987). In his study of electricity generation and distribution systems, Hughes showed that engineers, who are system builders, are able to relate together domains of life

different than the purely technological – John Law calls this “heterogeneous engineering”, of which Netscape open sourcing is an example. Software companies represent a specific example of this process, since it is not an individual inventor but the company that becomes the “heterogeneous engineer”. Commercial companies have to address, in parallel, commercial goals and technical issues. This occurs as interactions between managers and engineers take place. Lucy Suchman suggests that the later are “in an ambiguous position between their accountability within the terms provided by those who employ them, and the premise that their value to their employer lies in the autonomous exercise of their professional skill” (Suchman 2000b). Internal associations of power are being woven around software, which is both the product and the core of the socio-technical system.

ANT assumes that delegated non-human characters silently prescribe us social norms and take part in constituting social order (Latour 1992), which is “an effect generated by heterogeneous means” (Law 1992). Relations embodied in materialities last longer because these are more durable than thought or speech. Objects not only are durable: ordered through time, but allow for ordering through space, due to their mobility. What Bruno Latour calls immutable mobiles are objects whose “immobility within network space affords displacement within Cartesian space” (Law 2000a). When delegated to immutable mobiles, order can be enforced at a distance. What Law calls preferences of objects is a power effect resulting from durability. To explain this durability, ANT differentiates between an “open” network and a black box. Punctualization is a process of simplification in which a complex network first “acts as a single block, then it disappears, to be replaced by the action itself and the seemingly simple author of that action” (Law 1996). Felix Stalder quotes Callon’s and Latour’s concept of a black box, a sealed network whose contents need no longer to be considered. Naturally, the box can be reopened and its stability is tied to the costs of its opening. Black boxes lend stability to the networks in which they are incorporated. Technological artifacts often function as black boxes. From this perspective, software represents an unusual kind of materiality. Its durability is not material, it is rather an effect of distribution: the more widely adopted, the more it can enforce its preferences through standard-setting activity. Software is dependent upon a hardware layer and thus can be considered a materiality, but on its own it is ephemeral, comparable rather to laws and norms than material objects.

Software fits the description made by de Laet and Mol of a mutable mobile, an

object with “boundaries that are vague and moving rather than clear and fixed” (de Laet and Mol 2000) – but its story is slightly different than that of a Zimbabwean bush pump described by de Laet and Mol. The pump is a solid and mechanical, a simple tool made for use in non-complex settings that becomes mutable because it can easily adapt to different settings and circumstances. While immutable mobiles allowed control at a distance, a system of bush pumps can function at a distance even if the manufacturer does not retain control over the equipment. Software is a mutable mobile par excellence, a result of a *permanently beta* production process, with a family of versions being used simultaneously not in separate, discreet settings, but over one interconnected network. It is a temporal fluidity, with versions from different moments coexisting in one space. Internet is a network space in which localities are either distinguished through functional means - into the creator community, developer community and the mass of users - or by such technological factors as available bandwidth or access. Proprietary software is created centrally and even with F/OSS projects there is only a limited number of sites where the software is developed. Even in these conditions, fluidity of software is on one hand a result of market pressures and on the other of the interconnectedness afforded by the Internet, which weakens borders and distance between center and peripheries; and lowers costs as well. In the *permanently beta* mode of software production, developers retain control over software objects that are increasingly mutable and complex.

The Netscape case does not therefore fit Suchman’s observation that “stance of design from nowhere is closely tied to the goal of construing technical systems as commodities that can be stabilized and cut loose from the sites of their production long enough to be exported en masse to the sites of their use” (Suchman 2000b). Software, either proprietary or F/OSS, is not created “out of nowhere”: the producers are either in the media spotlights or visible online, the company is a powerful brand name and CEOs and developers strive to achieve renown in the community. Suchman (2000b), after Wagner, suggests that design communities are characterized by external detachment and internal intimacy. In the Netscape case, this detachment is not a norm. Zawinski’s critical comments about Netscape accused the company precisely of that: detaching itself from the demands of the market and the users, and considered this a failure on the part of the company.

All this suggests that software is an object with extremely limited independence from its human creators. It cannot depend upon material durability or spatial

distancing from its source to achieve independence. For an artifact such as software, Suchman's observation that "persistent presence of designers/users in technoscientific discourse [...] reflects a durable asymmetry among human and nonhuman actors" (Suchman 2000a) is especially true. It is people who "conceive and initiate technological projects, and configure material-semiotic networks" (Suchman 2000a), and "vitality and autonomy of computational artifacts emerge through the camouflaging of the networks that support them" (Doyle, quoted in Suchman 2000a). This statement is not a result of an a priori differentiation between humans and machines. Once created, software, provided that it is successful, achieves a form of durability as a standard-setting entity, even if it exists in the form of a family of versions. In this sense, computer code can be considered a regulator of the online space (Lessig 1999) – and increasingly of the offline space as well, as the networking of societies continues. But in the process of code creation or reconstruction software shows no preferences beyond structural limits of programming languages and path dependency of earlier versions, if any. It might be most appropriate to use Latour's concept of an actor network as a circulating entity "providing actants with their actions" (Latour 1997). Thus, over software's life cycle first the developers structure the software code, while later the binary acts upon the users. This is especially true with software such as a browser, which sets the general frame for navigating and viewing the online space.

The Missing Masses are at the Shadow Theater.

The multitude of Internet users plays a peculiar role in the Netscape story. Latour writes in an essay on the "new Princes", actors that gain social advantage by being able to understand and address both social and technological power, that consumers constitute a front in their struggle and the new Princes will go to extremities "in order to interest, please, seduce, force, capture, or imprison consumers" (Latour 1988). The *permanently beta* model of continuous redesign is also an attempt to fully integrate and cater to user responses. But if we look closer at the association that Netscape CEOs were forming, the end users play the role of only a statistical indicator: they are needed to provide a high market share figure for the new software. Furthermore, since this figure was dropping and Netscape was losing to Microsoft, it was not a factor that was crucial for the goal of winning over the server market. End user popularity would become important when the new client would ship, but initially the objective of increasing business customers' confidence

was to be achieved by anchoring the Mozilla project with external developers, and not by appealing to customers.

“Information wants to be free”, the slogan accredited to Stewart Brand and considered a “hacker credo” is a recurring motive in descriptions of the Internet. Implied in the slogan is a belief that by providing access to information, universal publishing ability and cheap, global-scale means of communication, Internet has a liberating function. The online space is often described using the metaphor of the Greek agora, a public market, festival and assembly space. As an online agora, Internet is an open space that has the potential of nurturing electronic democracy.

The Netscape case suggests that a different metaphor is more fitting, one that accounts for obstacles to free communication and universal participation. A Javanese shadow puppet play, the *wajang*, can serve as such metaphor. In the play, *Dalang*, the puppeteer, sits on one side of a wide screen on which puppets cast large shadows. “From the *dalang*’s side of the screen – where traditionally only the men were permitted to sit – one sees the puppets themselves, their shadows rising up dominant on the screen behind them; from the reverse side of the screen – where the women and children sit – one sees their shadows only” (Geertz 1973, 132). *Dalang* is a symbol of whom Latour (1998) calls the new Prince and who “like Proteus, is an individual, an assembly, a techno-structure, a nation or a collective” (Latour 1988). The screen represents an information filter, an effect of both spatial and temporal distance as well as technological and institutional filtering. On the other side of the screen sits the underprivileged mass with filtered access to information. Geertz writes that “the shadows of the puppets are identified with the outward behavior of man, the puppets themselves with his inward self” (Geertz 1973, 137). Generalizing this we can state that on the *dalang*’s side actors can view and access the inner workings of the network, while on the other side only black boxes are perceived and the scope for action is limited. Translation and punctualization of the network can only be conducted from one side of the screen.

In the case of Netscape, the screen separates the developer community, intimate with software and skilled enough to shape it, from the end users. The former is divided into a core group of hackers, “who enjoy exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary”⁵ and less capable programmers, still able not only to use, but also produce software. The end users are

⁵<http://catb.org/esr/jargon/html/H/hacker.html> (Accessed 01.10.2003)

seen by the hackers as those “using [the computer] as a means rather than an end”, “who use a program from the outside, however skillfully, without getting into the internals”⁶. To them, Netscape addressed a warning on the site, from which the Communicator code could be downloaded: “If you have never compiled and debugged a very large C/C++ program before, all that downloading the following will do for you is use up a lot of disk space,” (Oakes 1998d).

So the shadow theater screen represents a border that separates an intimate audience from a mass remaining in the dusk due to insufficient knowledge or data manipulation skills. In the Netscape case, the actual code is available to programmers, while end users only experience the shadow play of ready products, their branding and marketing. They can appreciate the return of the Mozilla mascot and use F/OSS, but while participating in the F/OSS system, they lack skills necessary for access to its technical core. Since end users cannot profit from freedoms afforded by F/OSS licenses, distinct software brands become an alternative incentive to use F/OSS. Brand names can punctualize networks of code in a similar manner as binaries. Slogans such as “Information wants to be free” or “Free the Lizard” obscure complex power relations present in F/OSS projects, suggesting that open sourcing is simply about freeing code, and – as a result – human users as well.

Josh Boyd (2002) suggests that the rhetoric of “community” is often used online despite the fact that we often are not dealing with classic communities. Boyd shows how in the case of eBay auction site this rhetoric is used to build user and consumer trust in an association that is more like a small country than a small community (Boyd 2002). In many online milieus, this rhetoric is used, or even a sense of community exists, while in fact stable differences in levels of involvement exist among “members” - for example between posters and lurkers in discussion groups. More importantly, the metaphor suggests that sometimes, as in the example of Netscape's F/OSS community, the division is due to a division in knowledge and skills, which cannot easily be bridged. The shadow theater metaphor is a way of restating Wagner's concept of “detached intimacy” of scientific and technical communities (quoted in Suchman 2000b). According to Suchman, technologists have a position of “detached intimacy” – detachment from the rest of the society, especially “sites of technologies-in-use” is coupled with intimate relations with their profession, company and technologies (Suchman 2000b). Detached intimacy is the “joint creation of an

⁶<http://catb.org/esr/jargon/html/U/user.html> (Accessed 01.10.2003)

elaborate social world within which one can be deeply engaged, but which remains largely self-referential, cut-off from others who might seriously challenge aspects of the community's practice" (Suchman 2000b).

Where are the effects?

The switch of Netscape Communicator code to F/OSS was an unusual innovation. It was a radical innovation at a time when a conservative one would be expected, due to technological momentum, understood as a sum of material, political, economic components that a socio-technical system acquires and that with time form a conservative force pushing the system along an earlier set trajectory (Hughes 1994, Hughes in: Bijker, Hughes and Pinch 1987) terms, . This can be explained by extreme innovativeness and adaptation to change that is characteristic of software companies in comparison with classical "brick and mortar" enterprises, as well as by the specific, much weaker form of durability of software technology. It was also unusual because network space actually embraces fluidity. John Law writes that fluidity of objects becomes visible and is considered a threat when network's failure leads to a panic (Law 2000b). The Netscape case is an example of a situation, in which fluidity of software was invited and openly admitted in a move to save the company. Netscape Company wanted to transform its revenue model and save itself as a commercial entity. F/OSS hackers wanted to jumpstart a F/OSS marketing campaign. Developers wanted to develop code they were intimate with.

Netscape company successfully interested and mobilised all actors who agreed that open-sourcing of Netscape code is an obligatory passage point for the whole network. But in the end the developers were not successfully mobilized. After a year, Mozilla project included 100 developers hired by Netscape and only about thirty outside developers, probably less by an order of magnitude than expected. Massive involvement and a tremendous amount of feature innovation anticipated by Andreessen did not occur (Oakes 1998a). Netscape could not therefore profit from the expertise of external developers, as was hoped by Zawinski. F/OSS failed as both a frame for collective work and a business model. Zawinski (1999) provides a couple explanations. The code was big, complicated and unfriendly, so that the necessary foundations, of which Behlendorf (1999) spoke, turned out not to be that strong. During the year Mozilla "released a large pile of interesting code, but it didn't much resemble something you could actually use" – the code could not be punctualized into a binary.

In order to function properly, networks have to be simplified, routinized, reduced to a “black box” with few but well defined parameters. F/OSS code, engaged in a *permanently beta* state, is unable to achieve precisely that. Apparently, even hackers that work with source code have limited ability of perceiving and manipulating networks of code. A successful project needs code that can be punctualized into binary because only then developers can test it and end-users can use it. One strategy is thus to “release early, release often”. Another, used by the Linux community, is to run in parallel a “stable” path for end-users and a “development” path in which new components are continuously integrated and innovation flourishes in the complexity of code (Tuomi 2001).

Even a bold move to install a new generation engine at the heart of the browser, which set back development schedule and meant difficulties as far as competing for customers went, did not draw the attention of developers. One year after the release of Netscape code, Jamie Zawinski resigned from his post at Mozilla. It took Mozilla three more years to release a beta version (0.9) of Mozilla browser in May 2001 and one more to release Mozilla 1.0 in June 2002. Since then, Mozilla.org is again in the *permanently beta* state and regularly shipping out new versions and modules, although Mozilla developer community did not grow significantly. The troubled past of the browser is covered up and a “history” page on the Mozilla site creates an impression of gradual development over years. A changed developer crew, a merger of Netscape with AOL and newly reworked major components of the software mean that in four years only the brand names in fact remained the same. Netscape has used the F/OSS code to ship Netscape Navigator 7, but the new version has small chance of successfully competing with Microsoft, whose Explorer controls about 90% of the market.

But it would not be true to call the Netscape open-sourcing a failure. Netscape successfully transitioned from a secretive development model and for a year was running a large software project in the open. Though the developer community did not participate actively, it provided much needed feedback. A number of useful development tools was released and the process did play a key role in marketing the “open source” concept by providing a major commercial actor that could initially legitimize it.

The F/OSS code was successful in the intellectual rights regime. Netscape code was freed from commercial attachments, confirming once again that not specific code, but the general legal framework is the greatest success of F/OSS. The

case of Netscape suggests that F/OSS politics achieve similar successes as the environmental movement described by Dieter Rucht (1999). Rucht shows that although the state of environment has worsened, the movement itself is a success as an “agenda setter”. Open sourced Netscape code with time proved to be a strong association: a “self-sustaining network that is able to resist dissociation” (Law 1987, 114). It was able to enforce its preference that the code remains free. From a F/OSS perspective, commercial actors are spokesmen who, when saying what software wants and needs, displace software’s preferences by speaking in the language of proprietary rights. Microsoft, the anti-hero of the software universe, is such only in a historical and contingent manner. Potentially, every company that dreams of setting own standards (and Netscape clearly wanted to achieve that in 1994) runs the same risk. F/OSS is a remedy to this, since it ensures that no punctualization into proprietary form is possible.

Where are the creators?

De Laet and Mol describe Dr Peter Morgan, the inventor of the bush pump, as a “non-classical hero, who makes no claims to heroic actorship” (de Laet and Mol 2000). They suggest that “sometimes abandoning control may contribute to spreading what one has been making” (de Laet and Mol 2000). The engineer from Zimbabwe could abandon control because the simple pump can be transformed locally and operates in an environment that is safe, mainly in terms of market competition. Complexity of software means that only experts are able to transform it and the commercial market space, in which software exists, is a hostile environment, with a constant flow of new competing projects and market expectations. The successful inventor has to constantly nurture and improve his project. Distribution possibilities provided by the Internet coupled with the norm of strong client support mean that shipped products never travel far, in terms of network relations, from the inventor.

In the Netscape case, Jamie Zawinski presents himself as being most intimate with the code. He is truly concerned with its quality and future and is among the three developers that formally moved from Netscape to the Mozilla.org structure. Zawinski was heterogeneously engineering the Mozilla development and in the year following March 1998 not only wrote code and ran the Mozilla organization, but “spent much time striving to convince people that mozilla.org is not netscape.com” (Zawinski 1999). And Zawinski, like Raymond, understood “that what we needed to

mount was in effect a *marketing campaign* - and that it would require marketing techniques” (Raymond 1999) and knew that on the Web he could leverage his voice with that of corporations. While Morgan dissolved his actorship, Zawinski in crucial moments stepped into the spotlights. In November 1998, the takeover of Netscape by AOL made future of the Mozilla project unclear. Zawinski published at that time an open letter titled “Fear and loathing on the merger trail”, in which he stated that AOL may stop funding Netscape’s contribution to the project, but in the end “mozilla.org is not Netscape. And it is not now, nor will it ever be, AOL” (Zawinski 1998). According to “Wired”, this letter influenced AOL’s decision to continue financing the Mozilla project (Oakes 1999).

The bush pump story reads surprisingly close to any description of F/OSS ideology: “granting the pump’s ownership to ‘the people’ contributes to its success [...] they do not pay for the right to use it [...]the people have access to affordable technology” (de Laet and Mol 2000). Hopefully, I have shown that it would be an oversimplification to compare Netscape to an individual inventor, as in this case the new Prince was a complex collective of actors, a heterogeneous network woven of components that “pertain [to it] by one side only, but through the other sides, they escape from the world they constitute” (Latour 2001) and strive to attain other goals. Netscape was a commercial corporation and a technological innovator whose CEOs and developers built a “seamless web” of heterogeneous actors around a process of code open-sourcing. It is a punctualized brand name, a great shadow cast upon a screen which obscures Netscape’s internal and heterogeneous complexity and plays a part in a play about the open-sourcing of Netscape code.

F/OSS Movement as a Social Movement

F/OSS software and community is a network of heterogeneous elements, in which differing politics of computer code, attitudes towards the commercial software sector and to the software market mix or even clash. While the concept of a social movement seems best suited to describe this process, loosely grouping a range of actors, the Netscape case forces us to reconsider a simple view of F/OSS movement as a collective actor striving to liberate humans and code.

If F/OSS is a social movement, then it can be fitted in a wider network of global-scale protest movements and politics of code can be considered an issue that has a significance for the society at large and not just for hackers. Otherwise, shape and characteristics of software become a marginal issue that is significant only in their

hermetic milieu of programmers. Outside this milieu, F/OSS would be mainly an economic activity and its cultural and social promise would have to be treated as another online myth.

I believe that we can consider F/OSS a social movement, although an unusual one. While F/OSS movement is structurally similar to other existing social movements, its specificity is a result of the specificity of software as a materiality and means of protest; and of the specific role played by the mass of supporters.

According to the most general definition, social movement is a “uninstitutionalized group of unrepresented constituents engaged in sequences of contentious interaction with elites or opponents” (Tarrow 1998). First modern social movements were class movements, usual labour unions interacting against national states. In the 1980s, new social movements emerged, for which cultural and to a smaller extent political stakes, and not economic issues were important. For Alaine Touraine (1981) social movements that strived for cultural transformation were an important actor in the move from industrial to post-industrial societies. Alberto Melucci defines social movement as a type of collective action that is 1. a phenomenon involving solidarity and not a simple aggregation, 2. involving conflict and not consensus, and 3. breaching limits of compatibility of the system and thus requiring social change. F/OSS movement fits this definition although the limits of the system that are breached by F/OSS or – to use Touraine's terminology – the stake of the conflict is a specific one: software. Software is seen as an unusual stake of social and political action because technological factors are reduced to social one in public discourse. Both politics of technology and their effect upon broader politics is therefore difficult to grasp.

One way of understanding this is provided by Lawrence Lessig's concept of code as a regulator of the online space, comparable to the legal system. In a network society saturated with communication and computer technologies, in which division into the online and offline is ever less applicable, code begins to regulate the whole of social life. For this reason hackers have the potential of becoming lawmakers of sorts, code is becoming a means of exerting power for national states and the existence of a social, and not computing, movement becomes possible, for which control over code's characteristics becomes a political stake.

There is no doubt that hackers tied to the FSF, who openly declare that coding is a political activity, although outside the frame of traditional politics, form a social movement. But by including the “open source” segment as well we can see, that

F/OSS structurally reembles other social movements, in which both a radical, anti-commercial minority and a pragmatic mainstream, willing to co-opt commercial actors and be itself co-opted, strive to attain the same stake.

F/OSS as a social movement can be considered part of a wider global movement, for which global corporations, and not national state, become the key enemy. This global movement faces the difficulty of creating adequate forms of protest that will allow the movement to exert influence upon powerful commercial actors. Use of the human body as a tool of protest and symbolic conflicts conducted with the use of media are among the most widely used solutions. Specificity of computer code places F/OSS movement in a privileged position, since relatively low costs of software production allow hackers to create viable alternatives to proprietary products, instead of just pressuring corporations or market regulators to introduce changes. Software production is an unusual form of protest and for this reason F/OSS has to be treated as a hybrid of a social movement and market sector, since its main form of protest can also be considered a type of economic activity. In the end, the F/OSS sector is distinguished by much higher cooperation and solidarity than among ordinary economic actors in the market.

Alain Touraine assumes that each movement fights for control over historicity, understood as “control of the great cultural orientations by which a society's environmental relationships are normatively organised” (Touraine 1981, 26). This assumption is most difficult to prove in the case of F/OSS movement, since hackers have developed a hermetic culture, focused solely on computer code. Attempts made by Richard Stallman to equal free code with freedom of speech did not gain wide acceptance in the F/OSS community. Even if hacker ethic can gain wider acceptance, as Pekka Himanen (2001) suggests in stating that hackers are in fact cracking the “iron cage of modern rationality”, only a narrow group of specialists will still be able to interact and shape code. This is the difficulty shown by the Netscape case, which suggests that a division exists inside the F/OSS movement between hackers and end users.

In order to both enter the desktop software market and become a wider social movement, F/OSS movement must mobilize and involve end users as well. Use of F/OSS is even more difficult to justify in moral terms than its coding. Paradoxically, end users can therefore become involved in the F/OSS movement only as consumers, expressing their support through selection of software. Individual consumer choices become a key form of activity for the movement, as the ability to

protect open online standards and block attempts at market domination by proprietary software depends upon a large market share of F/OSS. We can compare F/OSS movement to other social movement organizations built on a “top-down” model. Greenpeace is a good example of a small organization of professional activists supported by a wide group of members who solely pay their member duties. In the case of F/OSS, the core of most involved hackers is supported both by a wider programmers' community and end users. To further complicate the issue, choice of F/OSS by the most important customers – who are not individuals, but companies, governments and state institutions - is dictated solely by pragmatic reasons. Use of F/OSS by these unexpected movement members is ambiguous and the software in itself is as “stupid” as the network, since it is a platform that supports any sort of socio-technical and value systems. While researchers observe that many social movements are most successful in transforming public discourse and, above all, members themselves, F/OSS seems able first of all to change the shape of the software market. Social changes become only a secondary effect that comes about through poorly understood socio-technical dynamics.

The F/OSS movement also fits in a network of other “open” projects, all of which function largely due to Internet's existence and are tied by a common belief in collective methods of knowledge collection and processing and a consideration of open knowledge as a key social value and common good. Here we can mention such projects as the Creative Commons licensing project, peer-to-peer networks, distributed computing projects, or initiatives such as MIT OpenCourseWare or Budapest Open Access Initiative. This wider movement, an *open information movement* understands the importance of cultural stakes and can be considered an example of Latour's “new Prince”, able to tie code and cultural content, and defending common cultural goods at the level of both computer code, legal regulations and cultural practices. F/OSS movement plays the role of an important, technologically-intimate segment of this movement, which as a whole can be seen as an important social actor fighting for control of historicity today.

Bibliography

- Andreessen, Mark, 1993, NCSA Mosaic for X 0.10 available, URL: <http://groups.google.com/groups?selm=MARCA.93Mar14225600%40wintermute.ncsa.uiuc.edu>, Accessed: 01.10.2003
- Behlendorf, Brian, Open Source as a Business Strategy, in: Mark Stone, Sam Ockman and Chris deBona (Eds.), Open Source. Voices From the Open Source Revolution, Sebastopol, Ca: O'Reilly and Associates, URL: <http://www.oreilly.com/catalog/opensources/book/brian.html>, Accessed: 01.10.2003
- Bijker, Wijbe, Thomas Hughes and Trevor Pinch (eds.), 1987, The Social Construction of Technology, Cambridge, Mass, MIT Press
- Boyd, Josh, 2002, In Community We Trust: Online Security Communication at eBay, URL: <http://www.ascusc.org/jcmc/vol7/issue3/boyd.html#References>, Accessed: 01.10.2003
- Callon, Michel, *Actor-Network Theory - The Market Test*, <http://www.comp.lancs.ac.uk/sociology/stscallon1.html>, Accessed: 01.10.2003
- 1999, Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of Saint Brieuc Bay, in: Mario Biagioli (ed.), The Science Studies Reader, New York and London: Routledge, pp.67-83
- Court, Randolph, 1998a, Navigator Free-for-All, "Wired", 22 January 1998, URL: <http://www.wired.com/news/business/0,1367,9816,00.html>, Accessed: 01.10.2003
- 1998b, Netscape Layoffs Begin, "Wired", 13 January 1998, URL: <http://www.wired.com/news/business/0,1367,9637,00.html>, Accessed: 01.10.2003
- de Laet, Marianne and Annemarie Mol, 2000, The Zimbabwe Bush Pump: Mechanics of a Fluid Technology, "Social Studies of Science", 30: pp.225-263
- FSF, 1996, The Free Software Definition, URL: <http://www.gnu.org/philosophy/free-sw.html>, Accessed: 01.10.2003
- 1998, Why "Free Software" is better than "Open Source", URL: <http://www.gnu.org/philosophy/free-software-for-freedom.html>, Accessed: 01.10.2003
- Geertz, Clifford, 1973, Ethos, World View, and the Analysis of Sacred Symbols, in: Clifford Geertz, The Interpretation of Cultures, Basic Books
- Glave, James, 1998, Netscape Shepherds Mozilla to the Masses, "Wired", 23 February 1998, URL: <http://www.wired.com/news/technology/0,1282,10486,00.html>, Accessed: 01.10.2003
- Hamerly, Jim and Tom Paquin with Susan Walton, 1999, Freeing the Source. The Story of Mozilla, in: Mark Stone, Sam Ockman and Chris deBona (Eds.), Open Source. Voices From the Open Source Revolution, Sebastopol, Ca: O'Reilly and Associates, URL: <http://www.oreilly.com/catalog/opensources/book/netrev.html>, Accessed: 01.10.2003
- Hecker, Frank, 2000, Setting Up Shop: The Business of Open-Source Software, URL: <http://www.hecker.org/writings/setting-up-shop.html>, Accessed: 01.10.2003
- Himanen, Pekka, 2001, The Hacker Ethic and the Spirit of the Information Age, Secker & Warburg
- Hughes, Thomas P., 1994, Technological Momentum, in: Merritt Roe Smith and Leo Marx (eds.), Does Technology Drive History? The Dilemma of Technological Determinism, Cambridge, MA, MIT Press, 1994, pp. 101-13.
- Lancashire, David, 2001, The Fading Altruism of Open Source Development, "First Monday", vol. 6, no. 12 (December 2001), URL: http://firstmonday.org/issues/issue6_12/lancashire/index.html, Accessed:

01.10.2003

Latour, Bruno, 2001, Gabriel Tarde and the End of the Social, <http://www.ensmp.fr/~latour/Articles/82-TARDE.html>, Accessed: 01.10.2003

----- 1988, How to Write "The Prince" for Machines as Well as for Machinations, URL: <http://www.ensmp.fr/~latour/articles/article/036.html>, Accessed: 01.10.2003

----- 1997, *On Recalling ANT*, <http://www.comp.lancs.ac.uk/sociology/stslatour1.html>, Accessed: 01.10.2003

----- 1992, Where are the Missing Masses? Sociology of a Door, URL:

<http://www.ensmp.fr/~latour/Articles/50-DOOR%20CLOSER.html>, Accessed: 01.10.2003

Law, John, 2000a, 'Ladbroke Grove, or How to Think about Failing Systems, published by the Centre for Science Studies and the Department of Sociology, Lancaster University, URL: <http://www.comp.lancs.ac.uk/sociology/soc055jl.html> (version: paddington5.doc, 15th August, 2000), Accessed: 01.10.2003

----- 1992, Notes on the Theory of the Actor Network: Ordering, Strategy and Heterogeneity, published by the Centre for Science Studies and the Department of Sociology, Lancaster University, URL: <http://www.comp.lancs.ac.uk/sociology/soc054jl.html>, Accessed: 01.10.2003

----- Objects, Spaces, Others (draft), 2000b, published by the Centre for Science Studies and the Department of Sociology, Lancaster University, URL:

<http://www.comp.lancs.ac.uk/sociology/soc027jl.html>, Accessed: 01.10.2003

----- 1987, Technology and heterogeneous engineering: the case of portuguese expansion, in: Wijbe Bijker, Thomas Hughes and Trevor Pinch (eds.), 1987, *The Social Construction of Technology*, Cambridge, Mass, MIT Press

----- 1996, *Traduction/Trahison: Notes On ANT*, published by the Department of Sociology Lancaster University, URL: <http://www.lancaster.ac.uk/sociology/stslaw2.html>, Accessed: 01.10.2003

Law, John and Annemarie Mol, 2000a, And if the Global Were Small and Non-Coherent? Method, Complexity and the Baroque, published by the Centre for Science Studies and the Department of Sociology, Lancaster University, URL: <http://www.comp.lancs.ac.uk/sociology/soc096jl.html> (version: nokia4; 28th January, 2002), Accessed: 01.10.2003

----- 2000b, Situating Technoscience: an Inquiry into Spatialities (draft), published by the Centre for Science Studies and the Department of Sociology, Lancaster University, and the Department of Philosophy, the University of Twente, URL: <http://www.comp.lancs.ac.uk/sociology/soc052jl.html>, Accessed: 01.10.2003

Law, John and John Hassard (eds.), 1999, *Actor-Network Theory and After*, London, Routledge.

Lerner, Josh and Jean Tirole, 2000, *The Simple Economics of Open Source*, NBER Working Paper No. w7600, URL: <http://papers.nber.org/papers/w7600>, Accessed: 01.10.2003

Lessig, Lawrence, 1999, *Open Code and Open Societies. Values of Internet Governance*, URL: <http://cyber.law.harvard.edu/works/lessig/final.pdf>, Accessed: 01.10.2003

Mendelson, Haim, *Netscape Case. Management in an Information Age*, URL: <http://pages.stern.nyu.edu/~tucci/netscape/home.htm>, Accessed: 01.10.2003

Neff, Gina and David Stark, 2002, *Permanently Beta: Responsive Organization in the Internet Era*, URL: http://www.coi.columbia.edu/pdf/neff_stark_pb.pdf, Accessed: 01.10.2003

Oakes, Chris, 1998a, *Developers On Free Netscape Code: Follow Through*, "Wired", 23 January 1998, URL: <http://www.wired.com/news/technology/0,1282,9860,00.html>, Accessed: 01.10.2003

----- 1998b, *Feeding Frenzy Over Netscape Code*, "Wired", 1 April 1998, URL: <http://www.wired.com/news/technology/0,1282,11378,00.html>, Accessed: 01.10.2003

----- 1998c, *Mozilla.org: We Are Not Netscape*, "Wired", 24 November 1998, URL: <http://www.wired.com/news/business/0,1367,16477,00.html>, Accessed: 01.10.2003

----- 1998d, *Mozilla Source, Revision 1*, "Wired", 10 April 1998, URL: <http://www.wired.com/news/technology/0,1282,11594,00.html>, Accessed: 01.10.2003

----- 1998e, *Netscape Drafts Free Code License*, "Wired", 5 March 1998, URL: <http://www.wired.com/news/technology/0,1282,10741,00.html>, Accessed: 01.10.2003

----- 1998f, *Netscape Frees Communicator 5.0 Code*, "Wired", 22 January 1998, URL: <http://www.wired.com/news/technology/0,1282,9813,00.html>, Accessed: 01.10.2003

----- 1998g, *Netscape Rolls the Dice*, "Wired", 31 March 1998, URL: <http://www.wired.com/news/technology/0,1282,11358,00.html>, Accessed: 01.10.2003

- 1999, Where's Netscape's New Browser?, "Wired", 18 March 1999, URL: <http://www.wired.com/news/technology/0,1282,18549,00.html>, Accessed: 01.10.2003
- Oakes, Chris and Craig Bicknell, 1998, Mozilla Stomps Ahead Under AOL, "Wired", 24 November 1998, URL: <http://www.wired.com/news/business/0,1367,16466,00.html>, Accessed: 01.10.2003
- Open Source Initiative, 2003, The Open Source Definition, URL: <http://www.opensource.org/docs/definition.php>, Accessed: 01.10.2003
- Perens, Bruce, 1999, The Open Source Definition, in: Mark Stone, Sam Ockman and Chris deBona (Eds.), Open Source. Voices From the Open Source Revolution, Sebastopol, Ca: O'Reilly and Associates, URL: <http://www.oreilly.com/catalog/opensources/book/perens.html>, Accessed: 01.10.2003
- Raymond, Eric S., 1998a, Cathedral and Bazaar, URL: <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/>, Accessed: 01.10.2003
- 1998b, Goodbye, "free software"; hello, "open source", URL: <http://www.tuxedo.org/~esr/open-source.html>
- 1999, The Revenge of the Hackers, in: Mark Stone, Sam Ockman and Chris deBona (Eds.), Open Source. Voices From the Open Source Revolution, Sebastopol, Ca: O'Reilly and Associates, URL: <http://www.oreilly.com/catalog/opensources/book/raymond2.html>, Accessed: 01.10.2003
- Rein, Lisa, 1998, Netscape: Bring on the Frankenbrowsers, "Wired", 13 February 1998, URL: <http://www.wired.com/news/technology/0,1282,10264,00.html>, Accessed: 01.10.2003
- Rheingold, Howard, 2002, Smart Mobs. The Next Social Revolution, Cambridge, MA, Perseus Books
- Rucht, Dieter, 1999, The Impact of Environmental Movements in Western Societies, w: Marco Guigni, Doug Mcadam i Charles Tilly (red.), How Social Movements Matter, University of Minnesota Press
- Stalder, Felix, 1997, Actor-Network-Theory and Communication Networks: Toward Convergence, URL: http://felix.openflows.org/html/Network_Theory.html, Accessed: 01.10.2003
- Stallman, Richard, 1985, GNU Manifesto, URL: <http://www.gnu.org/gnu/manifesto.html>, Accessed: 01.10.2003
- Stutz, Michael, 1998, Freed Software Winning Support, Making Waves, "Wired", 30 January 1998, URL: <http://www.wired.com/news/technology/0,1282,9966,00.html>, Accessed: 01.10.2003
- Suchman, Lucy, 2000a, Human/Machine Reconsidered, published by the Department of Sociology, Lancaster University, URL: <http://www.comp.lancs.ac.uk/sociology/soc040ls.html>, Accessed: 01.10.2003
- 2000b, Located Accountabilities in Technology Production, published by the Department of Sociology, Lancaster University, URL: <http://www.comp.lancs.ac.uk/sociology/soc039ls.html>, Accessed: 01.10.2003
- Tarrow, Sidney, 1998, Power in Movement. Social Movements and Contentious Politics, Cambridge University Press
- Touraine, Alain, 1981, The Voice and the Eye, Cambridge
- Tuomi, Ilkka, 2001, Internet, Innovation, and Open Source: Actors in the Network, "First Monday", vol.6, no. 1 (January 2001), URL: http://firstmonday.org/issues/issue6_1/tuomi/index.html, Accessed: 01.10.2003
- Veen, Jeffrey, 1998, Netscape Bares All, "Wired", 3 March 1998, URL: <http://www.wired.com/news/technology/0,1282,10633,00.html>, Accessed: 01.10.2003
- Viseu, Ana, 2000, How Social is the Social? Rethinking the role of artifacts in cognitive science, URL: http://fcis.oise.utoronto.ca/~aviseu/eng_art-cog_content.html, Accessed: 01.10.2003

Weber, Steven, 2000, The Political Economy of Open Source Software, BRIE Working Paper 140, URL: <http://e-economy.berkeley.edu/publications/wp/wp140.pdf>, Accessed: 01.10.2003

Winner, Langdon, 1985,1999, Do Artifacts Have Politics?, in: Donald Mackenzie and Judy Wajcman (eds.), The Social Shaping of Technology, 2nd Edition, Buckingham and Philadelphia: Open University Press, pp28-40

Wired News Report, 1998a, It's a Done Deal, "Wired", 24 November 1998, URL: <http://www.wired.com/news/business/0,1367,16459,00.html>, Accessed: 01.10.2003

----- 1998b, Netscape's Brain Transplant, "Wired", 10 November 1998, URL: <http://www.wired.com/news/technology/0,1282,16163,00.html>, Accessed: 01.10.2003

----- 1998c, Netscape Moves Community Front and Center, "Wired", 08 January 1998, URL: <http://www.wired.com/news/business/0,1367,9554,00.html> , Accessed: 01.10.2003

Working Group on Libre Software, 2000, Free Software / Open Source: Information Society Opportunities for Europe?, URL: <http://eu.conecta.it/paper/paper.html>, Accessed: 01.10.2003

Zawinski, Jamie, 1998, Fear and Loathing on the Merger Trail, URL: <http://www.mozilla.org/fear.html>

----- 1999, Resignation and Postmortem, URL: <http://www.jwz.org/gruntle/nomo.html>, Accessed: 01.10.2003