

Karsten Weber

Social Aspects of Non-Proprietary Software

Abstract:

This paper shall give a brief history of Free Software and Open Source, then shall describe the background philosophy of these social movements and the social aspects of the non-proprietary software community in more detail, and address possible problems which could arise, for instance, for public funding of non-proprietary software development or for concepts of responsibility in the ethics of technology. Finally, a possible future of non-proprietary software shall be outlined, which strongly differs from the bright forecasts of the mainstream Open Source and Free Software supporters.

Background philosophy

History and social aspects

Organization, responsibility and support

A possible future of non-proprietary software

Conclusion

Author(s):

PD Dr. Karsten Weber:

- Organization and contact address: Europa-Universität Viadrina, Postfach 17 86, 15207 Frankfurt (Oder), Germany
- Telephone, email and personal homepage: ☎ + 49 - 335 - 55 34 22 47 , ✉ kweber@euv-frankfurt-o.de, 🌐 www.phil.euv-frankfurt-o.de, 🌐 <http://www.phil.euv-frankfurt-o.de/extern/akv/>
- Relevant publications:
 - Informationelle Grundversorgung und Eingriffsfreiheit. Der Zugang zu Informationen aus der Perspektive politischer Philosophie. Habilitationsschrift, in Vorbereitung.
 - Philosophische Grundlagen und mögliche Entwicklungen der Open Source- und Free Software-Bewegung. In: Bernd Lutterbeck, Robert Gehring (Hrsg.): Open Source Jahrbuch 2004. Zwischen Softwareentwicklung und Gesellschaftsmodell. Berlin: Lehmanns Media, 2004.
 - Non-proprietary Software und Geschenkökonomie: Lösungen für die digitale Spaltung? In: Rupert M. Scheule, Rafael Capurro, Thomas Hausmanninger (Hrsg.): Vernetzt gespalten. Der Digital Divide in ethischer Perspektive. München: Fink, 2004.

Particularly, the following discussion of the background philosophies or ideologies of non-proprietary software must be understood as a reconstruction. It is plausible to identify the underlying ideas of Open Source and Free Software with libertarian as well as with communitarian positions within the political and social philosophy. But this doesn't mean that all supporters of this kind of software would agree to that reconstruction or would be conscious of these backgrounds. I shall not give an introduction of the historical development of libertarianism or communitarian ideas but shall try to identify the central ideas and their connections to non-proprietary software from a systematic point of view.

Due to several reasons, the discussion of possible developments of non-proprietary software has the character of unsafe forecasts. Despite all enthusiasm of supporters of non-proprietary software its social consequences have to be described as marginal. Yet, the social movement behind non-proprietary software cannot compete – with respect to its social importance – with, for instance, the human rights, anti-globalization, or environmental protection movements. One can also have doubts, whether the communities of Open Source and Free Software could be compared with respect to their goals. Richard M. Stallman (2001), founder of the Free Software Foundation, reproaches the ideas behind Open Source for taking technological and not social aspects into account in first line.

Taking into account the broad discussion, statements on non-proprietary software always are in the danger to mutate to declarations of personal convictions to be for or against such software and its background philosophy. The objectivity of statements on non-proprietary software is limited, the discussion is often characterized by loyalty or opposition and debates seldom are free of emotions. All that has to be taken into account if one wants to reconstruct the background ideas of non-proprietary software.

Background philosophy

The background philosophies – one perhaps can say "ideology", too – of Open Source and Free Software are different, particularly with respect to ownership of software. Eric S. Raymond stresses that Open Source means to have a non-fundamentalist attitude to the development of software. From his point of view, one can find a lot of pragmatic reasons, which are based on technological considerations, why the

source code of software should be open to all (Raymond, 2001: 117). Richard M. Stallman however, founder of the Free Software Foundation, insists that software should be free due to moral obligations to others and to the society we belong to (e. g. Stallman, 1992). But regardless of this antagonism of pragmatic and more ideological arguments the outcomes of Open Source and Free Software are quite the same. So, many authors coined names like "libre software" (Robles, Scheider, Tretkowski, Weber, 2001: 1) or "non-proprietary software" (Renn, 1998) to indicate both the differences and those aspects that Open Source and Free Software have in common.

Primarily Stallman and Raymond speak publicly in philosophical and ideological terms. Both describe themselves as libertarians, but their understanding about what it means to be a libertarian differs significantly. Stallman stresses that it should not be allowed that someone claims property rights on information or knowledge. From his point of view software is a kind of information or knowledge and so he claims that "information wants to be free" (Stallman, 1992). It seems that Stallman tries to argue in a left-libertarian Lockean style. Locke stressed that it is morally forbidden to take possession of natural resources like water or food without leaving enough to other people. If someone gets the exclusive right of disposal on such natural resources this would lead to a lack of resources of all other people. Now one – and Stallman did – can apply this idea to information and knowledge: It is morally forbidden to claim an exclusive right of disposal on ideas, information, knowledge, or software, because this will bring harm to all other people. Actually, one can find several authors who argue that way, in Germany for instance, Helmut F. Spinner (1994; 2002) or Volker Grassmuck (2002). Both argue that treating information and knowledge as a common good will produce more public benefits than exclusive usage by single persons, companies, or institutions; exclusive usage may even lead to a damage or defect. Here, they adopt Robert K. Merton's so called CUDOS model of scientific knowledge.

However, one can try to legitimate non-proprietary software by arguing another way. Within the libertarian philosophy private property is the most important right. Property rights are viewed as unconstrained: In case someone acquired a good legally, that is by taking possession of an abandoned good or by voluntary and fair exchange, nobody else is allowed to intervene in one's property rights. Applied to information, knowledge, or

software, this means that if a person acquires a computer program, she also acquires the exclusive right to do everything she ever wants to do with that program. This includes selling, giving away, changing, reengineering, and so on. From this standpoint, a voluntary exchange of course could include the limitation of the rights that one acquires, too. Here, Stallman and Raymond have quite different opinions on the meaning of libertarian ideas. Raymond as a right-libertarian (2000a, S. 3ff.) clearly would accept that software is a marketable good and that its usage can be limited by license agreements – software can be proprietary. But Raymond stresses that in the end keeping software proprietary will be inefficient (Raymond, 2000b: 4); his decision to support and to produce non-proprietary software finally is based on pragmatic considerations (O'Reilly, 1999), not on decisions of general principle. However, Stallman and other authors, for instance Aaron M. Renn (1998), claim, that software – as well as ideas, information, and knowledge – in principle should be free, because free access to and free exchange of software is an act of cooperation and solidarity among people. They stress that without such a cooperative behavior societies could not really exist.

Stallman (2003a) emphasizes, that “free” doesn't mean “free of charge” at all. He particularly doesn't want to socialize all goods. However, some authors claim that non-proprietary software is a contribution to Marxist or communist ideas (cf. Söderberg, 2002). Notwithstanding all the differences Raymond und Stallman both agree in the rejection of copyright and patents for software. From their point of view copyright and patents are morally unjustified infringements of property rights (Stallman, 2001).

Despite the fact that leading persons like Stallman and Raymond stress libertarian ideas and so the interests and rights of individuals are focused, one will find ideas of social philosophy that in some way contradict this libertarian orientation. Anarchism, Marxism, and Communism already were mentioned.

Libertarismus and anarchism can be united – a good example is Noam Chomsky's anarchism-syndicalism. But it is quite unusual that within the ideas of the proponents of non-proprietary software communitarian and libertarian aspects come together. However, in his book “The Hacker Ethic and the Spirit of the Information Age” Pekka Himanen (2001) clearly addresses, although implicitly, communitarian ideas. He argues that the most important motivation to produce non-proprietary software is to gain social capital. Himanen also stresses that this public welfare

orientation is a kind of ethics; he calls it “hacker ethics”. Consequently, from his point of view, this moral stance is not limited to the production of software, but could be extended around other parts of life and work, for instance teaching at schools and universities. Not only Himanen talks about this public welfare orientation but one can find many statements concerning that position. Even the attitude of “us and them” – the use of a concept of an enemy, which is very common in non-proprietary software community – can be interpreted in a communitarian sense. It shall help to make identification easier to non-proprietary software. However, many aspects of the background philosophy or ideology of non-proprietary software remain contradictory: the rejection of authority and the personalization or even idolization of some leading persons, the potential conflict of making profit out of non-proprietary software and a public welfare orientation, the sometimes utopian conceptions of common goods, the contradiction of the search for individual freedom and self-determination on the one hand and the sometimes strange sectarian attitudes on the other.

Maybe the social movement of non-proprietary software could be compared to the early beginnings of environmental protection movement in the seventies and eighties – both include complex social relationships and its members are very heterogeneous with respect to motivation. The social movement of non-proprietary software isn't consolidated yet, it is characterized by partisan fights and is to be expected to go through a couple of transformations and splitting-offs.

Historical and social aspects

Since Richard Stallman has founded the GNU project and the Free Software Foundation (FSF) and at least since Open Source – especially LINUX – gains more and more users, economical success (Wheeler 2003) and public awareness, benefits and risks of non-proprietary software are widely discussed – in scientific as well as in public debates. Most times, those discussions are focused on the LINUX-WINDOWS antagonism, some times on non-proprietary software as part of a solution of the digital divide, and increasingly on the economical opportunities that particularly are revealed by LINUX and Open Source.

The development of non-proprietary software is a coordinated and globalized but not institutionalized process which is its main difference compared to

scientific knowledge production. Non-proprietary software is produced either by single persons or (sometimes large) groups of volunteers. In most cases one can find so called maintainers who coordinate the software production process but there are no responsible persons in a moral or legal sense. Often the concept of responsibility is completely rejected by members of the non-proprietary software community as well as the idea that the production of non-proprietary software should be institutionalized. Now the historical development shall be looked at.

In the early sixties computers became widespread especially at universities in the USA. In those times software had to be written by the users themselves – standard software was unknown, a software industry didn't exist. Those who wrote software shared their developments with others – while doing so they behave like scientists. Programs were distributed in source code because of two major reasons: first, the plurality of computer architectures made it necessary to adopt and modify software to make it run. Second and more important, software was treated like the results of scientific research. The source code was open to everybody who wanted to use, change, or develop it further. As in the process of scientific research, nobody had claims on the results except those of authorship and priority but as well as in the scientific realm nobody accepted something like responsibility for consequences that could arise from using the software. Programmers of software, like scientists, felt only to be bound to the duty of accurateness and precision.

The late seventies and eighties brought some major changes in the development of software. The availability of small computers – especially personal computers like the Apple II or IBM's PC – created a software market (see Campbell-Kelly, Aspray, 1996: 260). Software became proprietary and its source code – and so the know-how within the source code – no longer was open in the sense described above. The scientific-styled software *development* became a market-styled software *production* (a brief introduction of the history of Open Source Software can be found, for instance, in Feller/Fitzgerald 2002 or Pavlicek 2000).

Richard Stallman, today one of the most important persons in the community of developers and supporters of non-proprietary software, didn't want to be part of this commercialized software production and left the MIT in 1984, where he worked for several years in the AI Lab (see Stallman, 2003a). He started the GNU project, which

means that he wanted to create a free UNIX-like operating system, and founded the FSF. His aim was – and is – to develop free software. It is important to say that “free” software does not stand for “free of charge” but for software which could be used, changed, maintained and distributed by everyone. To protect this status, Stallman created the GNU Public License (GPL, see Stallmann, 2003b). Within this license, the rights of users of free software are formulated and everybody who wants to use Free Software has to accept it. Stallman and the proponents of Free Software claim that this license is enforceable by US-American copyright law and believe that it guarantees that Free Software could not be commodified (see Stallmann, 2003c). Currently, in Germany there is a discussion, whether this viewpoint is correct or not. In the Internet one can find an expert opinion that denies that the GPL really is enforceable (see Spindler, 2003, unfortunately only in German).

In contrast to the moral and ideological style Stallman uses to argue for Free Software the proponents of Open Source, especially Eric S. Raymond, are much more pragmatic. They argue that software whose source code is open to everybody is much more reliable, easier to maintain, and faster to correct. So from their point of view it is much more rational to produce, maintain and use such non-proprietary software. This is true, Raymond stresses, especially in case of software which is part of other products like computer hardware. But he accepts that some people want to produce proprietary software; on the other side he believes that the end of such software is near.

Raymond is socialized similarly like Stallman; he supports basic ideas of the FSF and has published programs under the GPL. The differences in ideological, philosophical, and technical arguments (see Raymond 2001, 67ff.) that both uses to support their positions had the result, that Raymond now supports the Open Source movement. It arose in 1998 as a reaction to the announcement of Netscape to release the source code of its Internet browser as open source (OSI, 2003).

Linus Torvalds, who initiated and still manages the development of LINUX, is one of the most famous persons of the Open Source scene. Like Raymond he thinks and argues pragmatically. From his point of view the benefits of the development of Open Source software are mostly technological; the social benefits are more or less side effects to him. He accepts that commercial and proprietary software production is justified in some cases – he himself works in the software industry.

Simultaneously he manages the development of LINUX, whose first versions he development since 1991. He sees himself as a kind of “figure-head” who represents LINUX in the public; in addition, he makes the final decisions on the LINUX development and decides which code will be integrated into a LINUX release version. In fact, the development process of LINUX is organized hierarchical, which contradicts many opinions about LINUX and Open Source. In many publications one can find statements that Open Source is a kind of grass-roots democracy. But that’s not true; one better compare it with a meritocracy in which those people decide who are accepted as experts in the respective field. Interestingly, some comments on the development of non-proprietary software stresses that a hierarchical organization is the only way to provide quality assurance and to prevent that non-proprietary software projects crack down if nobody feels responsible to make decisions.

An interesting aspect of non-proprietary software is that it is characterized by personalization, sense of mission, and opposition. For instance, Glyn Moody begins his book with remarks on the opposition to Microsoft and then continues with Torvald’s biography (Moody, 2001). Sam Williams turned over this sequence as he wrote his biography of Richard Stallman (Williams, 2002). Stallman *is* Free Software, Raymond *is* Open Source, and Torvalds *is* LINUX – at least in the public. Among others, this tendency to personalize positions finds its causes in those persons. The publications of Stallman, Raymond, and Torvalds attest their large ego (e. g. Raymond, 2001; Stallman, 2001; Torvalds, Diamond, 2001); particularly Eric S. Raymond gives the impression of an egomaniac who suffers from no self-doubt, Richard Stallman in turn shows a strong non-conformist attitude, and all three men seem to try to be no “flashy” manager but to correspond to the cliché of a hacker. Torvalds, Raymond, and Stallman seem to have something like charisma; at the same time, they are aware of their charismatic effects and specifically use them to promote their ideas. All three men want to spread a message and use the admiration of the Free Software and Open Source community for their own purposes.

Opposition is the connecting tie for all of the people belonging to the Open Source and Free Software community; without opposition to proprietary software and to the fully commercialized software industry and especially to Microsoft and Bill Gates one cannot understand the dynamics of the development of non-proprietary software. Microsoft and its operating system WINDOWS together with

all of its applications are the professed opponents or even enemies of the supporters and developers of non-proprietary software. Yet, it isn’t completely clear which motives of the developers of non-proprietary software to oppose Microsoft are dominant. The disapproval of the immense profits, the rejection of the – objectively and legally justified unfriendly – business strategies against competitors, the request to have insight into the know-how and the source code of software, the strange believe that Bill Gates and Microsoft plan a conspiracy to rule the world, or the conviction that information, ideas, and so software shall be free to access by everyone: all those motives are amalgamated. Many authors who wrote about non-proprietary software and who focus their arguments on moral and social aspects, stress that one shall oppose Microsoft’s business policy. In contrast, however, some other authors criticize this view. They believe that this ideologically grounded conflict could be a risk to non-proprietary software. In the case that this concept of an enemy of non-proprietary software should disappear they fear that the social movement of non-proprietary software will loose its cohesion (e. g. Eunice, 1998; Bezroukov, 1999).

Organization, responsibility and support

Despite all that, non-proprietary software is a great technological success (Wheeler, 2003). The number of users of non-proprietary software, especially of LINUX, grows continuously but actually it is much smaller than the number of users of Microsoft products and operating systems. However, there is some rumor in the software market, at least since IBM decided to heavily support LINUX with money and man-power – IBM, for instance, spent in the year 2001 a billion dollars for the development of non-proprietary software (see Robert, Schütz, 2001: 16). In addition, companies like RED HAT or SUSE are growing continuously. They make their profit with the distribution of Open Source software and primarily with services.

The production of non-proprietary software is coordinated via the Internet. Without it, such projects like LINUX or the APACHE web server could not exist, because development, support, and distribution of such huge projects needs fast, cheap, and asynchronous communication between those people who are involved in the project. Most times, non-proprietary software projects are built around a relative stable development team. Around this core one can find a (large) crowd of people who support

the development by testing the software by using it and who contribute smaller patches to correct software errors. Most projects have a so called maintainer – sometimes there are more than one – who represents the project to the outside and usually decides which code will be used in release versions. The position of a maintainer does not include the concept of responsibility. In fact, there is no person – or a group of persons – who will take on the responsibility for errors of the software, for negative consequences, or for the completion of software releases on schedule.

Writing, supporting, maintaining, or documenting non-proprietary software is a voluntary act. One can easily be part of the community of developers and one can leave this community easily as well. There are only weak ties between those who support non-proprietary software and only in those cases, in which non-proprietary software is developed in an institutional context, for example at universities or in companies, one can find a stable organization. This leads to a severe problem for state authorities or companies, which plan to use or to support non-proprietary software. It is difficult to fund projects without any stable organization or at least a contact person. Probably this is one of the reasons why despite all the enthusiasm of the Open Source and Free Software community many state authorities and companies still hesitate to use non-proprietary software. Its advantages – free access to the source code, a large community of developers, fast debugging, and so on – are leveled out by its disadvantages – uncertainty of further developments, unstable support, lack of responsible persons.

A possible future of non-proprietary software

But at present non-proprietary software has considerable support; this applies both to the developer community and to companies and state authorities (Quirós, González-Barahona, 2001: 7; on the European level see Esteban, 2001). However, this does not have to remain that way. The following scenario undoubtedly is negatively shaded; the future doesn't have to develop that way but it could. Anyway it isn't a law of nature that non-proprietary software will contribute to the advantage of all people for all times.

Non-proprietary software cannot be commodified by companies, state authorities or single persons. It is protected by licenses like the GNU Public License

(GPL) and comparable licenses. So the present stock of non-proprietary software will remain open in the future. Its existence is owed to a common aim that is strongly characterized by the opposition against a certain business model or even a particular company – and of course this is Microsoft. In addition, one can find some general ideological aspects: public welfare orientation – sometimes even Marxist positions (e.g. Söderbergh, 2002), a distinct view on liberty, the rejection of authority, or the search for new forms of cooperation. Both motivations, opposition and ideology, have made it possible to produce software of high quality that is used by millions of people and by an increasing number of companies and state authorities. But in the contradiction of the success of non-proprietary software and the motivations to produce it rests a great risk. More and more it seems that commercial and professional aims to support non-proprietary software move to the foreground (Fink, 2002; Moody, 2001; Young, Goldman Rohm, 1999). In case that a substantial part of the development of non-proprietary software is done by companies to maintain their interests, this could cause that the support of the world wide community of developers will fade away, because those people could have the impression that they support just that what they reject on emotional and ideological grounds. Now it would be possible that especially LINUX and non-proprietary software in general will share the fate of UNIX: New developments would only be done for the aims of companies and state authorities and non-proprietary software finally would be ... dead.

Conclusion

Non-proprietary software represents a technological and economical challenge to the existing software industry. But simultaneously the social movement which aroused around non-proprietary software has to be described as very heterogeneous both in its aims and in its ideological grounds. At present it is completely unclear in which way this movement will develop in the future. It is important for all those who have to decide about using or developing non-proprietary software to recognize that there is no guarantee for a positive future. Social movements aren't companies – they are hardly calculable in their behavior and in their alterations, since their grounds lie in ideological convictions which are more or less irrational or at least do not have the clarity of economic rationality. The self-determination of companies is one-dimensional: to obtain profit. But the self-determination of the people belonging to the non-proprietary software community is multi-

dimensional: for instance public welfare orientation, liberty, utopian world conceptions, and distance to authority, opposition, following, sense of mission, enthusiasm about and curiosity in technology. All these motives can be interpreted in various ways, are changeable, unsteady, and full of contradictions – and therefore their future development is hardly foreseeable. Despite all critical words, however, it remains to recognize that non-proprietary software with its philosophical and ideological convictions lying behind represents an important social challenge not only for companies but for our societies.

References

- Bezroukov, Nikolai: *Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism)*. In: *First Monday*, Vol. 4, No. 10, 1999, http://www.firstmonday.org/issues/issue4_10/bbezroukov/index.html, last visit 12/04/2003
- Campbell-Kelly, Martin/Aspray, William: *Computer*. New York, Basic Books 1996
- Esteban, Juan Jesús Muñoz: *European Initiatives Concerning the Use of Free Software in the Public Sector*. In: *Upgrade – The European Online Magazine for the IT Professional*, 11/6 2001, 36-40
- Eunice, Jonathan: *Beyond the Cathedral, Beyond the Bazaar*, 1998, <http://www.illuminata.com/cgi-local/pub.cgi?docid=cathedral>, last visit 12/04/2003
- Feller, Joseph; Fitzgerald, Brian: *Understanding Open Source Software Development*. London et al., Pearson Education Ltd. 2002
- Fink, Martin: *The Business and Economics of Linux and Open Source*. Upper Saddle River, New Jersey 2002
- Grassmuck, Volker: *Freie Software zwischen Privat- und Gemeineigentum*. Bonn, Bundeszentrale für politische Bildung 2002
- Himanen, Pekka: *The Hacker Ethic and the Spirit of Information Age*. London, Secker & Warburg 2001
- Moody, Glyn: *Rebel Code*. Cambridge, Massachusetts, Perseus Publishing 2001
- O'Reilly, T.: *Schlüsse aus der Open-Source-Software-Entwicklung*. In: *Telepolis*, 1999, http://www.heise.de/tp/deutsch/special/wos/64_33/1.html, last visit 06/26/2004
- OSI: *History of the OSI*, 2003, <http://opensource.org/docs/history.html>, last visit 12/04/2003
- Pavlicek, Russell C.: *Embracing Insanity. Open Source Software Development*. Indianapolis, Sams Publishing 2000.
- Quirós, Pedro de las Heras; González-Barahona, Jesús M.: *Free Software Today*. In: *Upgrade – The European Online Magazine for the IT Professional*, 11/6 2001, 4-11
- Raymond, Eric S.: *The Cathedral & The Bazaar*. Sebastopol et al., O'Reilly 2001
- Renn, Aaron M.: 'Free', 'Open Source', and Philosophies of Software Ownership, 1998, http://www.urbanophile.com/arenn/hacking/fsv_os.html, last visit 12/04/2003
- Robert, Gilbert; Schütz, Frédéric: *Should Business Adopt Free Software?* In: *Upgrade – The European Online Magazine for the IT Professional*, 11/6 2001, 12-19
- Robles, Gregorio; Scheider, Hendrik; Tretkowski, Ingo; Weber, Niels: *Who Is Doing It? A research on Libre Software developers*. Institut für Informatik und Gesellschaft, TU Berlin, 2001, <http://widi.berlios.de/paper/study.pdf>, last visit 08/22/2003
- Söderberg, Johan: *Copyleft vs. Copyright: A Marxist Critique*. In: *First Monday*, Vol. 7, No. 3, 2002, http://www.firstmonday.org/issues/issue7_3/soderberg/index.html, last visit 12/04/2003
- Spindler, Gerald: *Rechtsfragen der Open Source Software*. Herausgegeben vom Verband der Softwareindustrie Deutschlands e. V., 2003, http://www.vsi.de/inhalte/aktuell/studie_final_safe.pdf, last visit 12/04/2003
- Spinner, Helmut F.: *Die Wissensordnung*. Opladen, Leske+Budrich 1994
- Spinner, Helmut F.: *Wissenskommunismus für Wissenskapitalisten – Anachronismus oder Futurismus des Informationszeitalters?* In: Böhme, Harmut; Braun, Christina; Burckhardt, Martin (Hrsg.): *Die Politik der Maschine*, Band 5 der Interface-Reihe. Hamburg, Verlag Hans Bredow-Institut 2002, 295-324
- Stallman, Richard M.: *Why Software Should Be Free*, 1992, <http://www.fsf.org/philosophy/shouldbefree.html>, last visit 12/04/2003
- Stallman, Richard M.: *Free Software and Beyond*, 2001, <http://www.mikro.org/Events/OS/ref-texte/stallman.html>, last visit 12/04/2003
- Stallman, Richard M.: *The GNU Project*, 2003, <http://www.fsf.org/gnu/thegnuproject.html>, last visit 12/04/2003

Stallman, Richard M.: *GNU General Public License*, 2003, <http://www.fsf.org/licenses/gpl.html>, last visit 12/04/2003

Stallman, Richard M.: *Violations of the GPL, LGPL, and GFDL*, 2003, <http://www.fsf.org/licenses/gpl-violation.html>, last visit 12/04/2003

Torvalds, Linus; Diamond, David: *Just for Fun: The Story of an Accidental Revolutionary*. New York, HarperCollins Publishing 2001

Wheeler, David A.: *Why Open Source Software/Free Software (OSS/FS)? Look at the Numbers!*, 2003, http://www.dwheeler.com/oss_fs_why.html, last visit 08/29/2003

Williams, Sam: *Free as in Freedom. Richard Stallman's Crusade for Free Software*. Sebastopol et al., O'Reilly 2002

Young, Robert; Goldman Rohm, Wendy: *Under the Radar: How Red Hat Changed the Software Business – And Took Microsoft by Surprise*. Scottsdale/Arizona, Coriolis Group 1999

Proceedings of the symposium "Localizing the Internet. Ethical Issues in Intercultural Perspective" sponsored by Volkswagen*Stiftung*, 4-6 October 2004, Zentrum für Kunst und Medientechnologie (ZKM, Karlsruhe)