

### De-commodifying software? Open source software between business strategy and social movement

Holtgrewe, Ursula; Werle, Raymund

Veröffentlichungsversion / Published Version  
Zeitschriftenartikel / journal article

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:  
SSG Sozialwissenschaften, USB Köln

#### Empfohlene Zitierung / Suggested Citation:

Holtgrewe, U., & Werle, R. (2001). De-commodifying software? Open source software between business strategy and social movement. *Science Studies*, 14(2), 43-65. <https://nbn-resolving.org/urn:nbn:de:0168-ssoar-122626>

#### Nutzungsbedingungen:

Dieser Text wird unter einer Deposit-Lizenz (Keine Weiterverbreitung - keine Bearbeitung) zur Verfügung gestellt. Gewährt wird ein nicht exklusives, nicht übertragbares, persönliches und beschränktes Recht auf Nutzung dieses Dokuments. Dieses Dokument ist ausschließlich für den persönlichen, nicht-kommerziellen Gebrauch bestimmt. Auf sämtlichen Kopien dieses Dokuments müssen alle Urheberrechtshinweise und sonstigen Hinweise auf gesetzlichen Schutz beibehalten werden. Sie dürfen dieses Dokument nicht in irgendeiner Weise abändern, noch dürfen Sie dieses Dokument für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, aufführen, vertreiben oder anderweitig nutzen.

Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

#### Terms of use:

This document is made available under Deposit Licence (No Redistribution - no modifications). We grant a non-exclusive, non-transferable, individual and limited right to using this document. This document is solely intended for your personal, non-commercial use. All of the copies of this documents must retain all copyright information and other information regarding legal protection. You are not allowed to alter this document in any way, to copy it for public or commercial purposes, to exhibit the document in public, to perform, distribute or otherwise use the document in public.

By using this particular document, you accept the above-stated conditions of use.

## De-Commodifying Software?

### Open Source Software Between Business Strategy and Social Movement

*Ursula Holtgrewe & Raymund Werle*

Focusing on open source software the origin, development and organisation of a process of de-commodification is examined in an industry that usually relies on strong provisions to protect intellectual property. Open source denotes a cooperative and voluntary mode of software development cross-cutting organisational boundaries and transcending relations of market exchange. Starting with the Open Systems Movement in the late 1970s, which was driven by business strategic and industrial policy interests and complemented by a spirit of mutual support in professional communities, a social movement type of collective action has emerged which develops knowledge as a public good. Competent communities share the norms of the hacker culture and cooperate in informal relations challenging the boundaries between private and public goods. But the open source idea has also been transformed into a business strategy by companies who provide basic software products for free and make money with complementary products and services.

*Keywords:* open source software, social movement, intellectual property

Computer software has not always been a commodity, an industry and a market. In the era when mainframes prevailed, it was treated as a public good. Only with the diffusion of workstations and personal computers has software turned into a valuable private good which can be traded separately from hardware. Like other information goods, software is expensive to develop and cheap to distribute. This is why the industry relies typically on a variety of technical and legal

provisions in order to protect intellectual property and recover the investments in software development. Licences and patents as institutions supporting commodification are thus critical issues and contested terrain in the field.

There are limits to commodification. Early on, tendencies to avoid commodifying software could be observed. Norms of “openness” and “interconnectivity” have been tied to the development of the Internet in many ways – the Internet in

turn has enabled open and informal practices of cooperation and collaboration. Recently, open source/free software has become prominent as a cooperative and voluntary mode of software development cutting across organisational boundaries and transcending relations of market exchange.

In this paper, we analyse the role of open standards and systems in information and communication technology (ICT) and the development of open source/free software as instructive examples of the limitations of knowledge commodification.

Economic theory tells us that intellectual property rights (IPR) such as patents and copyrights are legal means to protect the knowledge of an innovator, assuming that knowledge by itself is not an exclusive good. Without legal (or other) protection, it is argued, the innovator would not enjoy the privilege of being able to exclusively harness the knowledge, and therefore individuals or corporations might not be willing to invest in innovations any more. Through IPRs others are excluded from using this knowledge, which means that a resource that could be copied and distributed at low cost is kept scarce. This creates a market value of knowledge in which an economic profit can be appropriated by the knowledge owner.

Modern societies, however, have also developed other institutional regimes of knowledge generation and innovation. In the academic and scientific context, knowledge has traditionally been developed as a public good. The academic/professional rules and rituals of establishing priority and authorship, of quoting and peer recognition encourage publication. Material incentives are (sometimes

hypocritically) disregarded while the open circulation, correction and self-correction of knowledge is promoted.

A more radical or socially transformative logic of innovation is pursued by social movements in this area. They do not just rely on interests, (in material rewards or reputation) but also on normative commitments. They value the “creativity of action” (Joas, 1996) and the norms of mutuality and sharing, and they seek to extend the social space assigned to their practices, communities and projects. Challenging established institutions, they create communities which promote social transformation and social innovation aimed at greater participation and democratisation (Honneth, 1995; Holtgrewe, 2001a; 2001b). We are going to show that open source/free software computer enthusiasm (hackerdom) brings these features of social movements to technological innovation.

These modes of innovation suggest that commodification of knowledge is not inevitable at all but that the boundaries between commodification and de-commodification are drawn in different ways. We are going to examine the origin, development and social organisation of the idea of openness in general and of open source/free software in particular. Activities that support open technology promote de-commodification. Such activities are driven by a variety of motives – not necessarily including an explicitly anti-commercial motive – at the level of the individual actors, and they are embedded in an institutional setting that provides opportunities and constraints to develop and implement public good type technology. We argue that the emergence of open source/free software must be seen in the multiple contexts of profes-

sional norms and practices, social movements promoting the accessibility of technology as a means of creative self-expression and political and commercial efforts to expand property rights protection and treat all kinds of software components as marketable commodities.

After an initial definition of open source/free software in contrast to both proprietary software and freeware/shareware, we analyse an early concept of openness in the context of standards in ICT, where it triggered limited de-commodification of technology in the development of new markets and business strategies. In contrast to this, open source/free software is a more radical variant of the idea of openness. Here, openness is taken beyond professional norms and contexts and tied to the normative commitments of social movements. After the characteristics of software development and the motives of developers to contribute are investigated, we explore the development of open source/free software through the phases of the initial social innovations and increasing popularisation and success. We then address the strategic interests of companies and other actors in openness, comparing open standards and open source/free software, and we finally conclude that open source/free software combines and sustains traditions and paradigms of innovation beyond the economic model but is precarious in the face of pressures towards commodification.

### **Open Source and Other Open Software**

Open source software is software developed in non-commercial, voluntary projects in which a number of develop-

ers (ranging from a couple of dozen to hundreds or even thousands of people) create, test, improve, document and maintain computer programs and modules of programs. This requires a program's source code to be accessible. Programs generally are written in higher programming languages and then translated into machine language (compiled). With proprietary software, typically, the compiled version is sold. Open source projects describe themselves as decidedly open in a social sense as well: everybody who is able to deliver a qualified contribution is invited to do so. Discussion takes place over public mailing lists and newsgroups.

Open source software can be used virtually free of charge, but not all software which is made available at no cost is open source. Many application programs are public domain software, which can be downloaded via the Internet for free even though they have been developed in a commercial environment. Yet such gratis software cannot be changed by the users because the source code is closed. There are also commercial programs where the source code is sold, such as commercial program libraries which are used in software development. Furthermore, openness does not always imply access to and use of software at no cost. Frequently it only means that access and use must not be denied arbitrarily or made impossible by levying unreasonably high charges for access or use.<sup>1</sup>

### **Openness in Perspective: Open Systems and Networks**

The idea of openness has been around since the early days of information and

communication technology (ICT). It serves to legitimise industrial policy, provides a rationale for business strategy and is part of professional ideology. This will be illustrated subsequently. We first look at the so-called Open Systems Movement and the emergence of open software standards. These developments did not generally challenge the conventional order of intellectual property rights protection. Firms with a weak market position opted for openness in order to attract allies who had also realised that they were too weak to survive if they relied on their closed technology. Openness in this context meant that firms relinquished property rights to certain technical specifications, which other firms were then allowed to share (cf. Iversen in this issue). The components remained proprietary; only the interface specifications were open. Open interface standards were therefore intended to allow technical components produced by different firms to interoperate in systems and networks. This in the context of early computer networks would help to achieve a "critical mass" of users and stimulate a self-reinforcing growth process. Thus intellectual property rights were not conceded in general. Rather, the boundaries between private property and public goods were redrawn when it appeared appropriate given the conditions of an emerging network economy (Shapiro & Varian, 1999).

Two elements of the development of computers triggered the openness argument in the area of ICT. The first was the evolving dominance of one single firm in the computer industry. After a period of competition for leadership in the market for commercial computers IBM introduced the System/360 mainframes in

1964. The architecture of this computer allowed upgrades to larger machines without rejecting existing software and thus secured backward compatibility. This was an extremely valuable asset because many programs and procedures running on the mainframes were written by system operators and could not be purchased on the market. A market for software effectively did not exist. IBM's machines attracted many new users, and the internal compatibility advantage locked them into IBM, which in the late 1970s had a share of about 70% of the market for mainframe computers (Grindley, 1990).

The second element was the emergence of computer networks and the difficulties of interconnecting computers of different brands or of linking different networks to one another. Pioneer users of networks in the 1970s were large corporations in the electronics and automobile industries, and firms in the financial services sector. The multinationals took the lead in creating private global networks, primarily to facilitate intra-organisational data exchange. The major computer vendors developed software that supported interconnection of their machines and consequently, the early data networks were proprietary. They were not compatible with other vendors' networks and often could not even integrate other vendors' machines. As a side effect, proprietary computer networking strengthened IBM's dominant position even more.

It was the competition with IBM which initially prompted European computer manufacturers to engage in activities towards open systems and open standards. They became apparent in several facets of the so-called Open

Systems Movement. In 1983, eight major manufacturers of information technology based in Europe set up the European Standards Promotion and Application Group (SPAG), which aimed at facilitating the implementation of standards issued as non-proprietary open standards by the International Standardisation Organisation (ISO) or the International Telecommunication Union (ITU). In 1987, SPAG initiated the formation of the European Workshop for Open Systems (EWOS), which was created by European associations of technology suppliers and user organisations and was meant to provide for collaboration on open systems. EWOS included European standardisation organisations and also, from the political side, the General Directorate IX of the European Commission (Werle, 2001a). Two other initiatives of the Open Systems Movement were linked to the operating system Unix and struggled to create a unifying Unix standard which was truly open and competitively neutral (Cool & Gabel, 1992): X/Open, an alliance of five European computer manufacturers, was forged in 1984 to counterattack IBM and other non-European vendors (Gabel, 1987) but after a few years came to include most major international computer vendors and IBM. The Open Software Foundation (OSF), set up in the USA in 1988, was a non-profit research and development organisation sponsored by many universities, in addition to computer manufacturers and software firms. At the time of the foundation of OSF, Unix was already split into several competing systems and had partly lost its attraction as an open system.

Other promoters of the idea of open systems in the 1970s and 1980s were the

operators of telecommunications networks. The majority were national public monopolies with an interest in expanding their telephony monopolies into the area of computer networks. They claimed to offer openness and provide world-wide connectivity similar to the telephone system by developing standards according to a layered model of computer networks. This model, the Open Systems Interconnection (OSI) frame of reference, was meant to provide a framework for open, i.e. non-proprietary, standards at a time when vendor-specific proprietary networks prevailed. Both international standardisation organisations, the ITU and the ISO, adopted OSI in the early 1980s. Not only the telecommunications operators but also many computer manufacturers and national governments committed themselves to OSI (Schmidt & Werle, 1998), which, similar to Unix for computer systems, provided a central reference point for the Open Systems Movement in the area of computer networks.

With their commitment to open standards of the OSI type, most European governments including the Commission of the European Union pursued two goals. They demonstrated their endorsement of the value of openness and at the same time used OSI as an instrument of industrial policy to the benefit of European computer manufacturers. The governments, directly or indirectly, controlled the public telecommunications operators and tried to use their procurement power to create a market for open computer and network technology in the sense that the standards, especially interface standards, were non-proprietary.

This picture of the Open Systems Movement as being guided by business

and industrial policy concerns is, however, incomplete. The role of the engineering profession, or parts of it, as a driving force towards openness needs to be added. One illustration may suffice. Early on in the area of data networks, computer scientists and engineers discussed strategies and technical options for a computer network which would interconnect all computers in the world. One forum for discussions and preparatory work was provided by the International Federation for Information Processing (IFIP), a federation of national technical and professional societies dealing with information processing. In the late 1970s, for instance, IFIP's Technical Committee 6 (data communication) discussed technical design issues in the field of electronic mail and message handling systems for computer networks. The organisation was particularly suited to this kind of voluntary preparatory work because it relied on a broad and heterogeneous membership of professionals from ICT corporations, network operators, research organisations and universities from different parts of the world, and was guided by the spirit of collaboration and exchange of ideas. From the outset, there was a consensus that it would not be desirable to base a system which was to provide global connectivity on proprietary technology. The contours of a completely new system were designed which was meant to complement and subsequently replace proprietary systems. The ideas were fed as pre-standardisation work into the official process of standard setting in the ITU (Schmidt & Werle, 1998).

The Open Systems Movement has shown that the openness argument was brought to bear by significant parts of

the technical community in order to overcome technical incompatibility barriers and realise the professional ideal of the global connectivity of computers and networks, which would also facilitate global collaboration. More important, however, was the fact that firms and governments promoted the openness argument to neutralise market dominance of a single firm and level the playing field. In line with this aim was the idea of using open systems and networks as universal platforms to which proprietary systems – including smaller ones which otherwise would not survive – could be connected. Benefits from network effects, which derive from the ability to interact with other systems and the availability of compatible complementary products, were thus to be spread more evenly.

The Open Systems Movement did not regard openness as a radical alternative to proprietary technology. It was not a social movement but a strategic alliance of actors in developing markets aiming to influence the rules of these markets – which, however, drew on professional norms and values as well.

### **Open Source/Free Software**

With the development of open source/free software, the professional norms and industrial strategies of openness were translated into a different context: from the adoption of standards to the actual collaborative development of software. Here, technological innovation increasingly became social innovation. With software for the Internet, there are no strict boundaries between standards, reference implementations and software packages. This is why a fair amount of open source/free software such as Apache

and Sendmail is involved in the running of web- and mail-servers. However, the best-known example of open source software is the operating system Linux.

### *Software Development, Hacker Culture and Diverse Economies*

The voluntary, collaborative and informal way of open source/free software development requires a look into the characteristics of software development in general and into the individual motivations of voluntary contributors.

Software development is distinct from other fields of research and development in several ways. The traditional view (and self-image) of software development as a craft or even an art (Knuth, 1972) has been corrected by evidence of its stratification and rationalisation in commercial industries (Beirne *et al.*, 1998). Yet, it still counts as a prime example of knowledge work, “a highly interpretive process” (Beirne *et al.*, 1998: 152) of creative problem-solving by applying, de- and recontextualising knowledge. This work, contingent and notoriously difficult to measure and control as it may be, results in “running code,” i.e. functioning software. Its combination of experience, situated knowledge and formalisation then results in a specific openness: software development has only partly been professionalised in so far that – in spite of the institutionalisation of computer science – the field is still open to autodidacts and people changing occupations. With the diffusion of PCs and the Internet, means of production are accessible in private homes as well.

Professional and organisational integration of software development is complemented by cultural practices and

norms shaping “communities of practice” (Wenger, 1998). They are focused on the culture of “hackerdom” (Levy, 1984; Himanen, 2001). Hackers in this sense are computer enthusiasts for whom programming is an expression of identity. Central normative orientations of hacker culture are the freedom of information and knowledge, universal accessibility of technology and a commitment to technological excellence and aesthetics (“elegant” code). This leads to a libertarian mistrust in hierarchies except for those based on merit. The boundaries between professional culture and hackerdom, then, are fluid. Hackerdom, however, adds characteristics of cultural *avant-gardes*, challenging established standards, and of social movements, seeking to transform institutions and social relations and to generalise their normative orientations.

While the actions of business firms and governments concerning open source/free software may be expected to follow rational calculations, the motivations of individual developers go beyond narrow economic rationality (cf. Weber, 2000). This phenomenon has been approached from different theoretical perspectives, drawing on theories of collective action and public goods, of gift and reputation economies, of intrinsic motivation and of social movements. Often, such theories are fairly close to central elements of the open source community’s self-perception.

According to Mancur Olson’s (1977) classic analysis of collective action, the production of collective goods is the less likely, the easier it is to free-ride, that is, to use the goods without making a contribution. In order to motivate actors to contribute to a collective good, selective



incentives are usually required that promise exclusive gratification to those who actively contribute. But if the public good is an innovation, the socially institutionalised recognition of inventors, founders and authors may provide an incentive sufficient to trigger initial contributions without any additional material reward. In this case the inventors are satisfied with the reputation they gain. If a minority of actors have a sufficiently high interest in the collective good, they will also invest in the good without needing (additional) selective incentives. After a “critical mass” of contributions is achieved the project will attract others to become involved (Kuwabara, 2000; also Marwell & Oliver, 1993), especially since software is a network good with positive network externalities.

Eric Raymond (1998; 1999), a prominent open source promoter, has argued that open source/free software projects function as gift economies, in which gifts are exchanged for reputation. The implicit rules of developer communities ensure the fair distribution of reputation, which in the case of Linux is assigned by central and recognised authorities who gratify contributors by including their contributions, if eligible, in new versions of the software. However, reputation is embedded in other, quasi-professional, norms that specify the requirements to be met by developers. Only functioning, well-written and elegant code is appreciated. Reputation cannot be pursued directly, therefore, but is gained as a diffuse reciprocal reward for the gift a developer gives to the community.

But investments in reputation may also be guided by economic motives. Since reputation may spill over from the

area of free projects into the sphere of paid work, successful contributions to open source/free software development can turn out to be rational investments in, notably, the careers of students of computer science. This is confirmed by a recent Internet survey of open source/free software developers (Robles *et al.*, 2001), which includes 5,333 respondents: only 19.6% of the developers have nothing to do with the software industry, 29.7% are students, 21% of the respondents are paid for developing open source/free software, and nearly half (45.5%) have profited from open source related activity in their careers, while another 26% are hoping to do so in the future. Yet 66% of the respondents spend less than 10 hours per week on developing open source/free software.

All this suggests that developers are fairly close to ICT professions, but strict commercialisation is limited. Also factors beyond network externalities, reputation, and career investments motivate individuals to participate in open source projects. These projects entail, as Rishab Aiyer Ghosh has suggested, intrinsic values such as beauty or challenge and a “fun factor”, which prevail in a “cooking-pot economy” (Ghosh, 1998). Limited individual contributions ensure access to a wide variety of contributions by others. The specific incentive for participation lies in using and enjoying this variety and in the opportunities of learning and pleasure. Free-riding is not a problem since the cost of distribution of the information goods is negligible – but free-riders miss out on the intrinsic gains of active participation. It should be noted, however, that individual contributions to open source/free software projects are more unequally distributed

than the normative idea of a “bazaar” (Raymond, 1998) of the free and open circulation of ideas and reputation suggests: among the 13,000 authors of the Linux kernel mailing list, who submitted some 175,000 contributions between July 1995 and April 2000, 2% had written more than half of the contributions (Moon & Sproull, 2000; cf. Ghosh & Prakash, 2000).

Most interpretations of individual motives for becoming involved in open source/free software are biased towards either straightforward economic factors or programmatic and philosophical elements. They ignore specific social contexts, structural and institutional constraints and opportunities, and the logic of the different open source/free software projects. Celebrating individual creativity and technological progress, the “philosophical” approaches reiterate the norms or “ideology” of hackerdom. This indeed provides a generic element of open source/free software. Yet understanding open source/free software requires more detailed sociological investigation. In the following sections we will examine the histories of key projects in order to identify relevant contexts and logics of development.

#### *Unix: Academic Communities and Telecommunications Engineering Traditions*

An account of the different facets of open source has to start with the operating system Unix, which, as we have pointed out, was already a focal reference point of the Open Systems Movement. Unix and the associated programming language C was developed by the Bell Labs, the research facility of the US telecom-

munications corporation AT&T. It was originally intended for in-house use by AT&T, but its significance for the computer industry soon became obvious. Work on this operating system had already started in 1965, and in 1972 the first version of Unix written in C was available. This facilitated portability of Unix from the beginning. It could operate on every computer which supported a C-compiler (Dunphy, 1991). As a result, machines from different vendors were compatible if Unix was implemented. These computers could be connected via telephone or data networks and exchange information. At the time Unix was developed, AT&T enjoyed a regulated telephone monopoly in the USA. However, regulation restricted AT&T to the telephone business and barred it from commercial activities in the computer industry including the provision of public data communication services.

This is a crucial reason why, in the early 1970s and for the next decade, AT&T granted Unix licences and made the source code available to universities and research organisations at a nominal fee. It was also the hope of AT&T that the implementation of Unix on many computers would stimulate computer networking and thereby generate traffic on the telephone lines. This expectation partly materialised after Unix-based software for the exchange of information between computers was developed (Leib & Werle, 1998). The Unix-to-Unix Copy Protocol (UUCP) provided the software backbone of Usenet, a system of newsgroups and bulletin boards which was later absorbed by the Internet (Hauben & Hauben, 1997).

In the 1970s Unix was freely shared with university researchers and develop-

ers. In this context, Dennis Ritchie, one of the Unix inventors, aimed to build not just an operating system and a programming environment, but also “a system around which a fellowship could form” (Ritchie, quoted in Grassmuck, 2000: 3). This fellowship was rooted in the traditions of academic research and publication and of telecommunications engineering with its established orientation towards interconnectivity. Consequently, Unix became a popular operating system among universities and laboratories that both used and developed the system.

Collaboration was close especially with computer scientists at Berkeley University, who in 1980 gained contracts from the Defense Advanced Research Projects Agency (DARPA) to modify Unix for computer networking (McKusick, 1999; Wayner, 2000). They improved AT&T’s Unix and built the Internet protocol stack TCP/IP on top of it. Their releases were distributed under a liberal licence which permitted the redistribution of both modified and unmodified code, in source code or binary versions, provided that the copyright in the source code was not changed and Berkeley was given credit in the documentation. Since the code AT&T had contributed to these distributions was not free, however, users still needed to license Unix from AT&T. TCP/IP in turn was distributed with AT&T’s Unix as well.

#### *Incipient Commercialisation and Resistance*

With the emerging commercial interest in software as a commodity, conflicts between commercial interest and the collaborative academic/engineering traditions arose and were constitutive for the

invention of open source/free software. Keeping up the traditional modes of collaboration required some social innovation.

In the 1980s, companies such as Microsoft, Borland, Lotus or Novell had begun to commercialise software and distribute binary versions. American universities were legally entitled to appropriate, patent and sell scientific and technological knowledge which arose from federally funded research (Etzkowitz, 1994). University labs thus set up strong links to private industry and spun off commercial companies. After AT&T’s divestiture in 1984 the corporation entered the computer market and expected its Unix Systems Laboratory to make a profit. AT&T required university collaborators to sign nondisclosure agreements and increased licence fees for the source code of their Unix. The commercial activities increasingly focused on the distribution of stable commercial releases as opposed to research and experimentation.

Academic developers/authors whose work had gone into the software frequently felt expropriated by having to license the basics of their own work in the case of AT&T and Berkeley (Wayner, 2000: 51) or by seeing their commitment to free sharing and access violated. This process of software commodification mobilised resistance. Developers started to build their own free Unix versions and they invented the institutional structures to protect and support these activities.

The roots of this resistance then were both the professional norms of academic computer science and the counter-cultural commitments of hackerdom. The commodification of software was perceived as a threat to the norms and practices of sharing information

and to the perquisites of Unix hackers' creativity – and this threat, in turn, caused developer communities to reflect on the institutional and social contexts of their practices.

In the 1980s, the prominent open source/free software projects of GNU and BSD represent contrasting rationales of a struggle against commodification, which we call *disentanglement* and *activism*. These carry associations of Albert Hirschmann's (1974) distinction between exit and voice, but with a specific emphasis on skill and innovation. In the social spheres of computer science and hackerdom, technical competence manifests itself as a strong norm of inventive problem-solving: The highly valued response to a problem, be it a deficient or faulty tool or application, is not just to identify the problem and complain about it, but to directly provide and circulate a solution. This norm of technological practice and creativity was now applied to the social and legal context as well. Open source/free software projects focused on both the rebuilding of proprietary software and the development of social innovations which facilitate collaborative development and prevent the results from commodification.

#### Disentangling Unix Code and Pioneering Informal Collaboration

In response to AT&T's strategy of software commodification the Berkeley Computer Science Research Group (BCSRG) began to explore ways of "liberating" their code from AT&T's licences (Wayner, 2000: 43ff.; cf. McKusick, 1999). They first distributed their TCP/IP stack separately under the Berkeley Software Distribution (BSD) licence. It was put on

the Internet and soon became the reference standard for the Internet. This increased interest in Berkeley's developments, which led the group to make a bigger effort. Keith Bostic "pioneered the technique of doing a mass net-based development effort" (McKusick, 1999) by mobilising colleagues, friends and acquaintances to rebuild the Unix utilities owned by AT&T. This opened up informal collaboration beyond small teams of professionals and academics.

The kernel of the operating system was extremely difficult to rebuild since it contained code of both AT&T and Berkeley. In a quasi-philological effort, developers built a database to disentangle the parts and then rewrote the AT&T-owned code. The result, the modestly named "Network Release 2", was a near complete Unix system. It was completed and ported to different hardware both by companies and free groups of developers. Eventually a variety of more or less incompatible Unix systems emerged (cf. Wayner, 2000: 49ff.).

#### Activism: Hacking Software and Maintaining Freedom

On the east coast of the USA, Richard Stallman, frustrated by the proliferation of restrictions on the free sharing of code at the Massachusetts Institute of Technology's Artificial Intelligence Lab, initiated the GNU (a recursive acronym for "GNU's not Unix") project and founded the Free Software Foundation in 1984 to create a free version of Unix. He developed several widely used programming tools and became a leading open source/free software activist. Most significant is his legal construction of a licence, the so-called *copyleft* (cf. <http://www.gnu.org>),

which was to guarantee that open source software could always be distributed freely. This “General Public License” (GPL) does not simply rule that products are free; it allows free distribution only on the condition that further developments and applications are placed under the same licence. Anyone who releases software that incorporates GPL-licensed code is compelled to use the GPL in the new release. The point of the GPL thus is its “infective” character. It is a tricky construction, which uses the legal instruments of copyright to subvert intellectual property. Copyright generally allows its holder to determine the conditions of the distribution and – up to a point – use of their products. Copyleft (“all rights *reversed*”) ties the product to the conditions of open use. This attempt to tie oneself and others to freedom and creative variety represents the social movement side of open source/free software, since it addresses questions of social transformation in the direction of freedom, learning, use-value and intelligent and cooperative use of products. Institutions, especially institutions of intellectual property hindering creative appropriation and development, are challenged and transformed.

The innovative and somewhat paradoxical construction of GPL copyleft has not become the standard for open source/free software communities. Rather, it has stimulated the proliferation of diverse licences developed by different influential projects, which secure openness but limit the infectiveness of the GPL (Perens, 1999). Open source licenses differ depending on the importance that is given to the following issues (Working Group, 2000: 8):

“Protection of openness. Some licences insist in that any redistributor maintains the same licence, and hence, recipient’s rights are the same, whether the software is received directly from the author, or from any intermediary part.

Protection of moral rights. In many countries, legislation protects some moral rights, like acknowledgement of authorship. Some licences also provide protection for these matters, making them immune to changes in local legislation.

Protection of some proprietary rights. In some cases, the “first author” (the party that originally made the piece of software) have some additional rights, which in some sense are a kind of “proprietary” rights.

Compatibility with proprietary licences. Some licences are designed so that they are completely incompatible with proprietary software. For instance, it can be forbidden to redistribute any software which is a result of a mix of software covered by the licence with any kind of proprietary software.

Compatibility with other open source licenses. Some open source licences are not compatible with each other, because the conditions of one cannot be fulfilled if the conditions imposed by the other are satisfied. In this case, it is usually impossible to mix software covered by those licenses in the same piece of software.”

The reasons for this are not just authors’ claims on intellectual property rights. The demands of the GPL tend to create problems as soon as open source developments meet proprietary software and hardware specifications. If, for instance, an interface of open software with a commercial program is to be developed

or hardware drivers are to be written, cooperation with commercial software firms is unavoidable because information about the proprietary systems is required. Even if the firms agree to cooperate they are unlikely to give up their own property rights.

#### Linux in the 1990s: Popularisation and Re-Entanglement

Open source/free software developments in the 1980s made use of the spaces for technological development under conditions of professional autonomy, which the Internet was based on – and which the technology in turn reinforced when open standards were established. When commercial interests emerged, the open source/free software community explicitly defended these spaces, the norms of scientific collaboration and publication and the public good character of their products. To do this, they drew on their own programming skills and their access to the means of production, i.e. computers and source code. Yet they also developed modes of cooperation through the Internet and rules such as the copyleft to legally protect open source. Thus in the 1980s open source was not a purely moral commitment, but chiefly remained the domain of professionals and experts.

In the 1990s, open source/free software spread and gained public attention beyond the computer labs and hacker communities. Linux is considered by some as a realistic alternative for everyday computer users (if they have friends skilled in computers), and the open source/free software mode of voluntary cooperation is currently being discussed both as a post-capitalist model for a lib-

erated society (Grassmuck, 2000) and as a paradigm of innovation and knowledge management (Moon & Sproull, 2000; Tuomi, 2001).

The success of Linux has taken place in an institutional context which is different from the 1980s. First personal computers and then Internet access have become affordable consumer goods for private households in industrialised countries. This makes global access to information and knowledge goods possible and facilitates communication and cooperation beyond well-equipped offices and labs. As a result, amateur expertise outside professional computing has developed, notwithstanding that standardised application software no longer requires users to have programming skills. The “hacker ethic” (Himanen, 2001) of technological expressiveness, creativity, global cooperation and sharing has spread beyond the world of professional computing to hobbyists of all kinds, net activists, and school children – together with and in spite of the frequently deplored commercialisation of content and standardisation of applications.

On the economic side, the expansion of the Internet has been part of fundamental changes in both hardware and software industries (Borrus & Zysman, 1997; Buss & Wittke, 2000). In general, innovation and value chains have been fragmented, moving from the vertically integrated systems manufacturers of the 1970s to a differentiated, more modular pattern. Hardware design, manufacture and assembly, software development for operating systems, standard applications, and user-specific adaptation, system integration and consulting services have been taken over by specialised companies. These compete and cooper-

ate both across and along value chains. We have “Wintelism” on the one side, denoting de facto monopolies in circumscribed sectors such as, notoriously, PC operating systems (Microsoft), as well as Internet routers (Cisco Systems) and fluid, networked structures on the other side, with firms cooperating, strategically merging, outsourcing and moving into neighbouring fields and out again (Naschold *et al.*, 1999). Governments tend to leave the provision of infrastructures that have traditionally been regarded as public goods to the market or to private sponsorship and public self-help schemes. Universities and research institutions find themselves competing for marketable innovations, and public funding of research and development is increasingly committed to deliver marketable results.

The institutional ensemble in the 1990s thus grants the market a larger space and the interactions and relationships of organisations and firms are more flexible and selective. Generally, incentives and pressures to commodify knowledge prevail. Yet in the field of software development the situation is ambiguous. Open source/free software is confronted with the tendencies to re-commodify software, but at the same time the advantages of de-commodified technology development are being discovered.

Originally, Linux was the kernel of a free, Unix-compatible operating system developed by Helsinki computer science student Linus Torvalds. Towards the end of 1990, he began to build a kernel for his 386 PC and looked for collaborators and found immediate support from interested communities of developers. The success of the project was based on several factors, three of them being suffi-

cient availability of PCs, useful documentation of Unix, and Minix, a small simplified Unix clone running on PCs. Minix had been developed to train students in operating systems and it had a large and lively user community. Torvalds released the source code of his operating system in autumn 1991. According to the norms of hacker culture, he put Linux under the GPL. It soon became the most prominent exemplar of decentralised collaborative development and involved up to 40,000 people under the “benevolent dictatorship” of Torvalds and a core team, who coordinated and maintained subprojects with quick and frequent releases which were just as quickly tested and improved.

In 1994 the first official Linux version (1.0) was released. After that, many new networks of developers entered the project and provided it with new opportunities and constraints. Support, together with commercial interest, came from hardware manufacturers who were interested in ports to their hardware. Frequently, developers in the companies promoted this interest. It was also in the early 1990s that the first companies were founded by Linux developers. These firms sell Linux distributions (Red Hat, SuSE and others), and/or offer software support, training and consulting services, customise software solutions based on open source or publish handbooks and magazines (West & Dedrick, 2001).

This shows that open source/free software is not necessarily incompatible with business interests. It is accepted as legitimate to make money with services related to open source/free software. Companies hire developers who continue working on free projects, grant hardware to open source projects and

sponsor programming camps and conferences. They also give up exclusive property rights to software in the interests of maintaining the public good, which has developed into an essential element of their business.

Also, in recent years foundations such as the Center for the Public Domain (<http://www.centerpd.org>) have been established, which provide funding and information to diverse free software groups, bring together academic experts, activists and entrepreneurs and oversee fair social exchange between the business and the non-profit community.

This does not mean that the relations between the commercial and the non-commercial sphere are always harmonious. There is an ongoing controversy over whether “freedom” is to be understood as a commitment to use-value and public goods or as freedom to participate in the market (cf. DiBona *et al.*, 1999). This in fact distinguishes the promoters of free software (Stallman and the GNU and Free Software Foundations) from open source (Eric Raymond, Bruce Perens and the Open Source Foundation). Tensions also arise from the diffusion of Linux products to lay users, both consumers and companies. They increasingly demand easy-to-use solutions, which directly compete with commercial software. Firms tend to tolerate resulting losses as long as they have the feeling that Microsoft is the firm to be hurt most by the open source products. However, the pressure on the non-profit communities will increase if and when commercial firms experience competition with open source products as a kind of expropriation through, for instance, the extension of copyleft to originally proprietary software.

### **The Interests and the Passions: Open Source/Free Software in Context**

Research and development in the area of computer software is a form of knowledge production which, like other areas of knowledge production, has increasingly become subject to commodification (David, 2000). This includes the devolution of intellectual property rights from individual academic researchers to institutions of higher education (Etzkowitz, 1994). The transformation of science from a public to a private good is controversial (Webster & Packer, 1997) and necessarily incomplete. Similarly, the commodification of software has evoked counter-reactions, with open source/free software being the most prominent one. But can it really be considered a radical challenge to commodification? The answer to this question is both no and yes. The balance tilts towards no if we look at the strategic use of open source/free software by firms and also by governments. It tilts towards yes if our focus is on the community of open source developers and on the social innovations connected with it.

Our examination of the Open Systems Movement has already shown that openness and de-commodification do not necessarily mean a transformation of the institutions of intellectual property. In this context, the openness argument was an “unconventional” one, that is to say, a new element of a business strategy (Garud & Kumaraswamy, 1993: 358) which gave up some property rights in order to open up new markets. Since the demise of the Open Systems Movement, firms have continued to use open strategies as a means of entering a market.



Confronted with open source/free software they are increasingly trying to develop "hybrid" licensing arrangements which make use of the possibilities of the open source/free software development models (external contributions to development), while seeking to avoid compromising intellectual property rights. Sun Microsystems, for instance, has been extremely successful using this strategy of relinquishing control in favour of adoption. Sun offers open sources strategically under the Sun Community Source License, which allows free access, experimentation and internal deployment, while retaining control over testing and distribution (Working Group, 2000: 9).

Unlike the conventional openness argument which originated in the Opens Systems Movement, the open source/free software idea challenges the boundaries of intellectual property rights, and indeed, companies experimenting with licences come under careful scrutiny *vis-à-vis* the definition of proper open or free licences (Perens, 1999). Yet in the late 1990s, several ITC firms such as Netscape and Apple started to release the source code for some of their products under "open" licences in order to receive input from external developers and to achieve compatibility of open source/free software with their own products and thereby improve these products. Today, HP, Compaq, Dell and even Intel and IBM also support or participate in open source projects (West & Dedrick, 2001).

The only big player in the ICT industry which explicitly counters open source/free software is Microsoft, whose prevailing position in the market for operating systems is being challenged by Linux. Microsoft has not only acknowledged the specific pressure being exerted by

Linux, but the firm has also conceded that open source/free software represents a long-term strategic threat because it is not compatible with the firm's licensing model ([www.opensource.org/halloween](http://www.opensource.org/halloween); Weber, 2000: 11). Supporting open source, such as Linux, represents a low-cost and low-risk attempt by hardware and software companies to re-establish competition and open up alternatives to cooperation with the monopolist (Winzerling, 2001). Again, commitments to openness emerge as the classical weak competitors' strategy intending to join forces and counterattack the market leader. Open source is increasingly viewed as a viable option to collectively developing software which can be used and sold complementary to or bundled with proprietary hardware or software products. Hardware vendors make commitments to open source/free software in order to make inroads into new market segments with their proprietary hardware on which open source code runs. Generally, firms are careful not to surrender the intellectual property rights of their own products. If they release source code, more often than not they attach provisions that assert their property rights to modifications of the original code or at least prevent others from claiming property rights to these modifications.

Yet, even the GPL and the other copy-left licences enable new business models based on open source/free software as a public good. Opening source code does not necessarily mean that it is distributed completely free of charge. If a distributing or developing company cannot commodify the software, the licences guarantee that competitors cannot either, and that improvements and modifications will be open as well.

Thus open source/free software strategies are not completely different from commercial concepts of customer orientation and collaborative development, which frequently require some sharing of knowledge between developers and users. Nor are they completely different from joint development efforts among competitors (Naschold *et al.*, 1999). They are, however, aimed not just at gaining market share and limiting development costs, but rather at “changing the rules in the information technology industry” (Working Group, 2000: 27). Firms have accommodated their business strategies to these tendencies in a variety of ways aiming at an advantageous co-existence of releasing and preserving property. This does not necessarily create a stable “equilibrium”. However, it has become less likely that intellectual property claims on the part of firms will trigger a process of “forking”, in which – as with Unix – a system branches into often proprietary, incompatible versions. Copyleft provisions of the GPL-type have so far succeeded to protect the public good nature of open source/free software. Yet, apart from this institutional solution, the firms’ accumulated experience of how to take advantage of the merits of open source/free software without compromising the intellectual property of core assets and, possibly, increased awareness of the role of public goods in knowledge production and innovation suggests that the history of Unix will not repeat itself with Linux.

Not only business firms but also governments are affected by and react to open source/free software. Generally, they take an ambiguous stance, although political interest is increasing. In the era of the Open Systems Movement,

European governments, in particular, promoted openness if it promised to serve industrial political goals. In the course of deregulation and liberalisation of the markets for information and communication technology and services, of European integration and of economic globalisation, the idea of openness was linked to a stimulation of competition both nationally and across borders. Telecommunications network operators were obliged to grant competitors access to their networks by disclosing interface specifications or implementing open interface standards. At the international level, governments committed themselves to open international standards. Diverging national standards and proprietary standards were identified as non-tariff barriers to trade (Werle, 2001b). Openness was expected to increase competition, stimulate innovation and boost economic welfare.

This situation is similar with open source/free software, which is also promoted by some governments to serve industrial policy goals. Recently, the European Union General Directorate for the Information Society initiated the creation of the Working Group on Libre Software, consisting of open source/free software developers from universities, research institutes and companies who have outlined the perspectives for open source/free software in Europe (<http://eu.conecta.it/>). Beyond the technical and cost advantages accruing from the extended use and intensified development of open source/free software, this group expects a renewal of competition in the software industry with competitive advantages for European companies. It suggests initial funding for promising projects, the support of testing fa-

cilities, and the use of open source/free software in public administration and education. Consequently, in the 7<sup>th</sup> call of the EU research programme “Information Society Technologies”, open source/free software projects are explicitly invited. While supporting open source/free software the EU stresses at the same time, in a Commission Green Paper on the patent system in Europe, that it is “vital to protect the fruits of innovation” through patenting because companies which sell patented products “have a competitive advantage when it comes to maintaining or expanding their market share.” Patents on computer programs and software-related inventions are explicitly included (European Commission, 2001: 1, 16).

The US government has traditionally supported open source/free software in universities and public research labs (Seiferth, 1999). Its cheapness accounts to a considerable degree for the rapid diffusion of the Internet in the US academic and research sector. However, it is also acknowledged that the Internet’s openness has stimulated creativity and innovation (CSTB, 2001: 107-150). Yet, the right of private firms to patent software that has been developed in the context of publicly supported projects has never been challenged by the US government (CSTB, 1999: 40-51).

Thus if governments are interested in open source/free software they eagerly try to avoid the impression that they agree with ideas of a fundamental transformation in the system of individual intellectual property rights (IPR). They acknowledge that IPR provide incentives to innovate but – when arguing in favour of open source/free software – they also stress that, while too little protection of intel-

lectual property prevents innovations, too much protection and the resulting “patent thicket” (Shapiro, 2001) may prevent the diffusion and recombination of knowledge.

Yet as we have seen, open source/free software is not simply the result of collective actors’ rational economic and political strategies of interest maximisation. It has been initiated and sustained by developers’ normative commitments to creative self-expression, sharing and informal collaboration which merge technological and social innovation.

Hackerdom may be seen as the ensemble of cultural and technological practices and normative motivations, which aim at innovative modes of informal collaborative software development and at new forms of licensing as tools for sustaining creativity. These motivations contain a logic of social movements addressing the transformation of the institutions of innovation. Innovation should not be exclusively associated with paid work and material gratification, but also regarded as an element of freedom, learning, and the intelligent and cooperative use of products which emphasises their use-value rather than their exchange-value. Institutions, especially those of traditional intellectual property rights, are viewed as hindering the creative appropriation and development of innovations and thus are challenged and transformed.

### **Conclusion: Open Source Between Commodification and De-Commodification**

The idea of openness and the normative commitments to this idea have their roots outside the commercial sphere. Different

traditions converge. They include a general understanding of academic, particularly basic research as a public good, a professional engineering view of telecommunications as a public infrastructure connecting people worldwide, and a conviction of computer scientists and engineers that open cooperation facilitates efficient and innovative use of software and hardware. They also include the interpretation of freedom as free access to computers, decentralisation of organisational structures and availability of space for creativity emphasised by the so-called hackers-communities of computer enthusiasts which comprise chiefly students and professionals of computer science and engineering, but also amateurs and autodidacts.

In the era of the Open Systems Movement the idea of openness was used predominantly to legitimise industrial policy and forge coalitions of business firms against monopolies. As openness was not intended to be anti-commercial, changing the traditional order of intellectual property rights was not an issue.

Hackerdom appears to be the additional element that has to be considered if we are to understand the way in which the traditional concept of openness was extended towards open source/free software. Open source/free software challenges the conventional wisdom that individual appropriability of the revenues on innovations is the institutional *conditio sine qua non* of an innovative and economically prosperous society. Such commodification may even be regarded as a threat to the wealth of a nation because it jeopardises long-term innovation by limiting access to knowledge and technology.

Open source/free software is by design

a process rather than a product. If source code is provided along with software packages or protocol stacks, these “products” can be easily modified and improved by users who at the same time are developers. Thus, it is difficult to protect individual intellectual property on pieces of open source software. As a social project open source/free software relies on the collaborative development of software and on sustaining the preconditions of such collaboration. Professional engineers cooperate with students of computer science, private industry exchanges pieces of software with public research and development units, hobbyists trade ideas with business people, political idealists who believe in the power of computers to change human life for better talk with realists who want to use machine power to make the status-quo more efficient. Frequently, people involved in open source/free software also play multiple roles themselves. They are, of course, users and developers of software, they invent new institutions of property rights such as copyleft and they develop the normative “theoretical” basis of the movement. They also move between multiple engagements in companies, networks and projects cross-cutting organisational and social boundaries. As “knowledgeable actors” they mediate between diverse organisational, professional and ideological (epistemic) communities (cf. Giddens, 1984). Who owns the results of such socially heterogeneous collaboration, such genuinely new modes of knowledge production?

Business firms avoid this question when they feel that participation in open source/free software can help to make inroads into new markets and challenge dominant firms. Likewise, governments

shy away from this question if they promote open source developments as cheap and flexible elements of a public knowledge infrastructure. Developers are not interested either if they get involved in open source to enjoy creativity and gain recognition and reputation.

Thus what evolves is neither public nor private, neither individual nor collective property. It is rather a fluid, hybrid mosaic of components, some of which are protected by copyright, others by copyleft and others again not protected at all. Each open source/free software project is precarious. It depends on a spirit of collaboration and diffuse reciprocity, as well as technological and institutional innovativeness and, to some extent, assurance against commodification. The latter is provided as long as the checks and balances of open source/free software work – these include a social movement which addresses the sustainability of creativity.

### Notes

- 1 Richard Stallman, one of the central promoters of open source/free software distinguishes between the meaning of “free beer” and “free speech”. In Europe the term “libre software” (French “libre” as opposed to “gratuit”) is used (Working Group, 2000).

### References

Beirne, M., Ramsay H. & Panteli A.  
1998 “Developments in Computing Work: Control and Contradiction in the Software Labour Process”. Pp. 142–162 in: Thompson, P. & Warhurst C. (eds.): *Workplaces of the Future*, Houndmills: Macmillan.

Borrus, M. & Zysman J.

1997 “Globalization With Borders: The Rise of Wintelism as the Future of Global Competition”. *Industry and Innovation*.

Buss, K.P. & Wittke, V.

2000 “Mikro-Chips für Massenmärkte – Innovationsstrategien der europäischen und amerikanischen Halbleiterhersteller in den 90er Jahren” (Micro chips for mass markets – innovation strategies of the European and US semiconductor producers). *SOFI-Mitteilungen* No. 28: 7-31.

European Commission

2001 Green Paper on the Community patent and the patent system in Europe. [http://europa.eu.int/comm/internal\\_market/en/intprop/indprop/8682en.pdf](http://europa.eu.int/comm/internal_market/en/intprop/indprop/8682en.pdf).

Cool, K. & Gabel, H.L.

1992 “Industry Restructuring Through Alliances: ‘Open Systems’ and the European Mainframe Computer Industry”. Pp. 357–380 in Cool, K., Neven D.J. & Walter I. (eds.), *European Industrial Restructuring in the 1990s*. Houndmills: Macmillan.

CSTB: Computer Science and Telecommunications Board, National Research Council  
1999 *Funding a Revolution. Government Support for Computing Research*, Washington, D.C., National Academy Press.

2001 *The Internet’s Coming of Age*, Washington, D.C.: National Academy Press.

David, P.A.

2000 *A Tragedy of the Public Knowledge ‘Commons’? Global Science, Intellectual Property and the Digital Technology Boomerang*. SIEPR Discussion Paper No. 00-02. Stanford: Stanford Institute for Economic Policy Research.

DiBona, C., Ockman S. & Stone M. (eds.)

1999 *Open Sources: Voices from the Open Source Revolution*. Cambridge, MA: O’Reilly.

Dunphy, E.

1991 *The UNIX Industry: Evolution, Concepts, Architecture, Applications & Standards*. Wellesley, MA: QED Information Sciences.

- Etzkowitz, H.  
1994 "Knowledge as Property: The Massachusetts Institute of Technology and the Debate over Academic Patent Policy". *Minerva* 32 (4): 383–421.
- Gabel, H. L.  
1987 "Open Standards in the European Computer Industry: The Case of X/OPEN". Pp. 91–124 in H. Landis Gabel (ed.), *Product Standardization and Competitive Strategy*. Amsterdam: North-Holland.
- Garud, R. & Kumaraswamy A.  
1993 "Changing Competitive Dynamics in Network Industries: an Exploration of Sun Microsystems' Open Systems Strategy". *Strategic Management Journal* 14:351–369.
- Ghosh, R. A.  
1998 "Cooking Pot Markets: An Economic Model for the Trade in Free Goods and Services on the Internet", *First Monday* 3, [http://www.firstmonday.org/issues/issue3\\_3/ghosh/index.html](http://www.firstmonday.org/issues/issue3_3/ghosh/index.html).
- Ghosh, R. A. & Prakash, V. V.  
2000 *The Orbiten Free Software Survey 1st edition, May 2000*, <http://orbiten.org/ofss/01.htm>
- Giddens, A.  
1984 *The Constitution of Society: Outline of the Theory of Structuration*. Berkeley: University of California Press.
- Grassmuck, V.  
2000 *Open Source Betriebssystem für eine freiheitliche Gesellschaft. (Open source operating system for a free society)* Presentation Evangelische Akademie Tutzing.
- Grindley, P.  
1990 "Standards and the Open Systems Revolution in the Computer Industry". Pp. 99–110 in Berg, J.L. & Schumny, H. (eds.), *An Analysis of the Information Technology Standardization Process*. Amsterdam: Elsevier.
- Hauben, M. & Hauben R.  
1997 *Netizens. On the History and Impact of Usenet and the Internet*. Los Alamitos, CA: IEEE Computer Society Press.
- Himanen, P.  
2001 *The Hacker Ethic and the Spirit of the Information Age*. New York: Random House.
- Hirschmann, A. O.  
1974 *Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States*. Cambridge, MA: Harvard University Press.
- Holtgrewe, U.  
2001a "Kreativität als Norm – zum Erfolg verdammt? Open-Source- Software zwischen sozialer Bewegung und technischer Innovation" (creativity as a norm – open source between social movement and technological innovation). Pp. 399–424 in Allmendinger, J. (ed.): *Gute Gesellschaft? Konstruktionen sozialer Ordnungen*. Opladen: Leske & Budrich.
- 2001b *Recognition of Service and Authorship: Struggles in and around the Knowledge Economy*. Paper presented to the Conference: *Participation between Markets and Organizational Democracy*. Munich, February 2001.
- Honneth, A.  
1995 *The Struggle for Recognition: The Moral Grammar of Social Conflicts*. Cambridge: Polity Press.
- Joas, H.  
1996 *The Creativity of Action*. Cambridge: Polity Press.
- Knuth, D. E.  
1972 *The Art of Computer Programming*. Reading, Mass.: Addison-Wesley.
- Kuwabara, K.  
2000 "Linux: A Bazaar at the Edge of Chaos", *First Monday* 5, [http://www.firstmonday.org/issues/issue5\\_3/kuwabara/index.html](http://www.firstmonday.org/issues/issue5_3/kuwabara/index.html)
- Leib, V. & Werle, R.  
1998 "Computernetze als Infrastrukturen und Kommunikationsmedien der Wissenschaft" (computer networks as infrastructure and communication medium of science), *Rundfunk und Fernsehen* 46: 254–273.
- Levy, S.  
1984: *Hackers: Heroes of the Computer Revolution*. Garden City, NY: Doubleday.
- Marwell, G. & Oliver, P.  
1993 *The Critical Mass in Collective Action*. Cambridge, MA: Cambridge University Press.

- McKusick, M. K.  
1999 "Twenty Years of Berkeley Unix. From AT&T-Owned to Freely Redistributable". In: DiBona, C., Ockman S. & Stone, M. (eds.), *Open Sources: Voices from the Open Source Revolution*. Cambridge, MA: O'Reilly.
- Moon, J. Y. & Sproull, L.  
2000 "Essence of Distributed Work: The Case of the Linux Kernel", *First Monday* 5, [http://www.firstmonday.org/issues/issue5\\_11/moon/index.html](http://www.firstmonday.org/issues/issue5_11/moon/index.html)
- Naschold, F., Jürgens, U., Lippert, I. & Renneke, L.  
1999 *Vom chandlerianischen Unternehmensmodell zum Wintelismus? (From the Chandlerian enterprise model to Wintelism)*. FS II 99-204, Berlin: WZB.
- Olson, M.  
1977 *The Logic of Collective Action. Public Goods and the Theory of Groups*. Cambridge, MA: Harvard University Press.
- Perens, B.  
1999 "The Open Source Definition", In: DiBona, C., Ockman S. & Stone S. (eds.): *Open Sources: Voices from the Open Source Revolution*. Cambridge, MA: O'Reilly.
- Raymond, E.  
1998 "The Cathedral and the Bazaar", *First Monday* 3, [http://www.firstmonday.org/issues/issue3\\_3/raymond/index.html](http://www.firstmonday.org/issues/issue3_3/raymond/index.html)
- Raymond, E.  
1999 *Homesteading the Noosphere*. <http://www.tuxedo.org/~esr/writings/homesteading/homesteading.txt>, August.
- Robles, G., Scheider, H., Tretkowski, I. & Weber, N.  
2001 *Who is doing it? A research on Libre Software Developers*. Berlin: TU-Berlin, <http://ig.cs.tu-berlin.de/s2001/ir2/ergebnisse/OSE-study.pdf>
- Schmidt, S. & Werle, R.  
1998 *Coordinating Technology. Studies in the International Standardization of Telecommunications*. Cambridge, MA: MIT Press.
- Seiferth, J.C.  
1999 "Open Source and These United States". *Knowledge, Technology & Policy* 12 (3): 50-79.
- Shapiro, C. & Varian, H.R.  
1999 *Information Rules. A Strategic Guide to the Network Economy*. Boston, MA: Harvard Business School Press.
- Shapiro, C.  
2001 "Navigating the Patent Ticket: Cross Licenses, Patent Pools and Standard-Setting". University of California at Berkeley, forthcoming in Jaffe, A., Lerner, J. & Stern, S. (eds.), *Innovation Policy and the Economy, Volume I*, Cambridge, MA: MIT Press.
- Tuomi, I.  
2001 "Internet, Innovation, and Open Source: Actors in the Network". *First Monday* 6, [http://www.firstmonday.org/issues/issue6\\_1/tuomi/index.html](http://www.firstmonday.org/issues/issue6_1/tuomi/index.html)
- Wayner, P.  
2000 *Free For All. How Linux and the Free Software Movement Undercut the High-Tech Titans*. New York: Harper Business.
- Weber, S.  
2000 *The Political Economy of Open Source Software*. BRIE Working Paper 140, Berkeley, CA.
- Webster, A. & Packer K.  
1997 "When Worlds Collide: Patents in Public-Sector Research", Pp. 47-59 in Etzkowitz, H & Leydesdorff, L. (eds.), *Universities and the Global Knowledge Economy. A Triple Helix of University-Industry-Government Relations*. London: Pinter.
- Wenger, E.  
1998 *Communities of Practice. Learning, Meaning and Identity*, Cambridge: Cambridge University Press.
- Werle, R.  
2001a "Institutional Aspects of Standardization - jurisdictional conflicts and the choice of standardization organizations". *Journal of European Public Policy* 8 (3): 392-410.  
2001b *Standards and the International Telecommunications Regime*. HWWA Discussion Paper, Hamburg. Forthcoming.

West, J. & Dedrick, J.

2001 "Proprietary vs. Open Standards in the Network Era: An Examination of the Linux Phenomenon". Proceedings of the Hawai'i International Conference on System Sciences (HICSS-34), Maui, Hawaii: IEEE.

Winzerling, W.

2001 LINUX – Durchbruch für die Open Source Entwicklung? (Linux – Breakthrough of open source development) Presentation, Oekonux- Konferenz, Dortmund.

Working Group on Libre Software

2000 Free Software / Open Source: Information Society Opportunities for Europe? Version 1.2 <http://eu.conecta.it/paper.pdf>

Ursula Holtgrewe  
Institute of Sociology  
University of Duisburg, Germany  
uholtgrewe@addcom.de

Raymund Werle  
Max Planck Institute for the  
Study of Societies, Germany  
werle@mpi-fg-koeln.mpg.de