Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2002 Proceedings

European Conference on Information Systems (ECIS)

2002

Open Courseware vs. Open Source Software - A Critical Comparison

Stefan Baldi
Munich School of Business, stefan.baldi@eba-muenchen.de

Hauke Heier
European Business School, hauke.heier@ebs.edu

Fabian Stanzick
European Business School London, fabian.stanzick@ebs.edu

OPEN COURSEWARE VS. OPEN SOURCE SOFTWARE – A CRITICAL COMPARISON

Stefan Baldi[1]

Hauke Heier[2]

Fabian Stanzick[2]

[1]eba Munich School of Business
Landshuter Allee 8
D-80637 München

[2]European Business School
Schloss Reichartshausen
D-65375 Oestrich-Winkel
Germany

Stefan.Baldi@eba-muenchen.de, Hauke.Heier@ebs.edu, Fabian.Stanzick@ebs.edu

ABSTRACT

The Massachusetts Institute of Technology (MIT) recently announced its OpenCourseWare project, which will make all its courseware available on the Internet free of charge. This paper compares the OpenCourseWare project to the open source approach of software development. It contrasts and evaluates the nature of the products, the motives of the producers and users, the consequences on the market environment and the contribution to the scientific progress. Besides exploring the specific MIT case, another motivation of the paper is to stimulate the academic discussion about new regimes for dealing with digital courseware.

1 INTRODUCTION

In April 2001 the Massachusetts Institute of Technology (MIT) announced the OpenCourseWare project [RICHARDS 2001]. The basic idea is to make all MIT courseware available on the Internet free of charge starting in 2002. This move seems not obvious in times when education is seen as one of the big future markets, with online education taking a big share of it [LUTTERBECK 2001]. The choice of the project name suggests some conceptual relation to open source software, which gained public visibility with the success of the Linux operating system. In this paper we provide an answer to the question whether OpenCourseWare of MIT can be characterized as the "open source project of university teaching". For both concepts we start by giving a short historic overview and the basic definitions. We then look at the motives for contributing on an individual and institutional level as well as on the effects on the respective markets and the scientific community. The concluding comparison focuses on the nature of products, the motives, and the effects of the approaches. Finally some suggestions for further development are given.

2 OPEN SOURCE SOFTWARE

The concept of free software, delivered with the source code is a comparatively old one. The first computers were research tools and software was freely passed on to anyone who needed it. Only later, when computers reached the business world, did developers begin to restrict the rights to their

software and to charge fees for each copy [DIBONA et al. 1999]. In 1984 RICHARD STALLMAN – a former MIT employee – founded the Free Software Foundation. His aim was to promote and to defend the idea of free software. In order to protect the rights of the authors and the free software itself from becoming proprietary, he designed a set of rights, codified in the GNU General Public License (GPL) [FREE SOFTWARE FOUNDATION 1991].

2.1 Definition

Source code consists of the programming statements and instructions that constitute the logical structure of the program. While the program can be executed without the source code its internal working cannot be understood and it cannot be modified. Open source has been characterized in different ways, e.g. as a methodology for development, a new business model or a political movement [WEBER 2000]. The common denominator of most definitions of the term "open source" is the publication of the source code of the software. The codified Open Source Definition [OPEN SOURCE INITIATIVE 2001] includes many of STALLMAN'S ideas, and can be considered as a derivative of his work [DIBONA et al. 1999]. Its main points are:

- *Free redistribution*: The software can be redistributed without royalties or licensing fee to the author (article 1).
- *Source code*: The program must include source code or where some form of a product is not distributed with source code, there must be a way of obtaining the source code for no more than the cost of distribution (article 2).
- *Derived works*: The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software (article 3).

The Open Source Definition differs from the GPL in the sense, that GPL requires any software integrating GPL-licensed software to be released under the same terms while the Open Source Definition does not require this. This is a critical modification because it allows commercial software to make use of open source software and removes the network effect of GPL.

2.2 Motives

Literature gives many different motives for programmers and software companies producing software under open source licenses as well as for using software based on these licenses. Broadly, they can be divided into methodological, economical and political/ethical reasons.

First, from a developer's perspective the source code is the precondition to produce individual and customized solutions. Second, the methodological argument stresses, that widespread cooperation in the development of software produces better results than an isolated, protected approach. The rationale is that a larger group of developers will produce a more useful and bug-free product because more people are reviewing the code. The larger user base also contributes to the development through more bug reports.

There are economic reasons as well why programmers favor open source. The most important one is that they can build on existing code. The costs of a new solution can be lower than that of software made from scratch or built on commercial components. While standard economic theory holds that free-riding inhibits private investments in non-rival goods such as software [BESSEN 2001], several authors have shown that this can change if additional factors are considered: The economic value of open source software can be in for-profit services (e.g. integration projects, consulting activities) that are sold with it or the reputation gained by the programmers combined with interesting job offers. Giving away the basic product for free and with the source code speeds up the network effect and thus increases the number of users. Also non-economic rewards such as enjoyment in solving problems, expecting reciprocal action and the socializing and cultural aspects have been cited [HIPPEL 2001, WEBER 2000, HIMANEN 2001].

The political/ethical argument emphasizes everyone's right to have free access to knowledge. RICHARD STALLMAN has been one of the most important supporters of the political/ethical argument. For STALLMAN the source code is knowledge that should be shared and distributed [DIBONA et al. 1999]. This claim is based on the fact that the marginal cost of distribution of software is extremely low. Copying of software does not reduce its utility to other individuals and there is no limited supply of copies, so why shouldn't all have the best software [WAYNER 2000]?

2.3 Effects on the Software Market and the Scientific Community

The effects of open source on the software market are diverse. The existence of open source has not led to the end of propriety solutions. Nevertheless, open source solutions are gaining increasing popularity and have caused a fundamental change in the software business. More and more big companies – like IBM, Sun, and Netscape – contribute to the open source software development.

Open source has a direct and an indirect effect on the scientific community. The indirect contribution constitutes the fact that major parts of the Internet are supported by open source software. Therefore open source contributes to the worldwide exchange of ideas, thus indirectly to scientific progress.

There is also a direct effect of open source on the progress of computer science. If open source software includes its construction plans, anyone, who knows how to read them, can profit from the ideas and insights of the original programmer. A discourse about the construction and improvement of software is stimulated.

3 MIT OPENCOURSEWARE

The concept of MIT OpenCourseWare [MIT 2001a] was born from discussions of a study group close to MIT's Council on Educational Technology. The task of the group was to consider ways to use Internet technology to improve education within MIT as well as MIT's influence on education on a global scale. The group was composed of faculty and staff from MIT, and was assisted by external consultants. At first, MIT expected to leverage its course materials as a way of extending revenues [WIGGINS 2001]. In April 2001 MIT announced the plan to make its courseware freely available on the Internet. The project is going to start in fall 2002 with an initial 500 courses. Over a period of 10 years MIT expects over 2000 courses to be available online. It will include material such as lecture notes, course outlines, reading lists and assignments for virtually all MIT courses across the Institute's entire curriculum e.g. in architecture and planning, engineering, humanities, arts, social sciences and management. The funding needed for the first 27-month start-up and pilot phase is about US$ 12 million.

While the idea of putting individual course material online is a widespread practice, the systematic organization of freely available course material from different academic disciplines in a standardized, searchable archive on an institutional level is an innovative approach.

3.1 Definition

Courseware consists of all supporting digital material for academic courses, such as presentation slides, case studies and software for educational use. MIT defines OpenCourseWare as follows: "[…] MIT course materials that are used in the teaching of almost all undergraduate and graduate subjects available on the web, free of charge, to any user anywhere in the world. […] MIT OpenCourseWare will provide the content of, but is not a substitute for, an MIT education" [MIT 2001c]. Additional important characteristics are: the restriction to non-commercial purposes, the retention of MIT intellectual property ownership of most OpenCourseWare material and voluntary participation of MIT faculty. No credit is awarded and no degrees are granted through MIT OpenCourseWare. It's important to note that the educational material is courseware for existing classroom courses and not for a specially developed e-learning environment.

3.2 Motives

MIT's motives to create OpenCourseWare can be divided into two groups. One group represents the reasons why OpenCourseWare is beneficial to MIT in its way to run university teaching. The other considers the outside effects.

Internally OpenCourseWare will allow the centralization of all courses in one electronic place, enabling to search, to cross-reference and to see the relationship between the courses. OpenCourseWare will stimulate the institute's renewal of its knowledge property and constitutes a way to compete with itself. OpenCourseWare will make it possible for MIT faculty to concentrate even more on the actual process of teaching, on the interactions between faculty and students that are considered the real core of learning [RICHARDS 2001]. While the project is led by MIT administration, broad support by MIT faculty is reported [MIT 2001b]. However, also critical voices have been cited [WIGGINS 2001].

The desired external effects are also important: MIT plans to showcase the quality of its teaching and use OpenCourseWare to attract prospective students. OpenCourseWare is seen as a model for university dissemination of knowledge in the Internet age and considered as proof of "an incredible idealism at the heart of MIT's educational mission" [RICHARDS 2001]. MIT expects that other universities, particularly those in developing countries, will use the material to enhance their own curricula [WIGGINS 2001]. Reputation and status are vital parts of the institutional motivation, but also of importance on the individual level of a professor. While research capabilities have been visible outside the own institution for a long time (by means of publications and conferences), this is not true for pedagogical qualities. The availability of courseware might change that.

3.3 Effects on the Courseware Market and the Scientific Community

The effects of MIT's initiative on the courseware market cannot yet be analyzed with sufficient accuracy because none of the courseware is online up to now. It will not be possible for MIT to sell the courseware in electronic form, though. MIT does not expect the initiative to undercut interest in attending MIT or willingness to pay MIT's tuition rates [WIGGINS 2001]. If OpenCourseWare has any implications on the sale of courseware of other institutions is hard to predict at this stage. This is also highly dependent on the type and quality of the material provided.

It should also be highlighted that MIT's raw course material could be compared by visitors to more sophisticated and integrated e-learning sites. This could even harm MIT's reputation. As long as MIT is not engaging in free e-learning activities but just offers accompanying course material no negative impact on online learning companies and initiatives (e.g. Fathom.com or UNext) can be expected.

As many educational book publications evolve from course material, it may be difficult to distinguish courseware from textbooks in the long run. This in turn might influence the future role of academic publishers. While faculty members typically retain intellectual property rights for books they publish based on their own research, the question of copyright is still open for the published courseware.

OpenCourseWare may have some positive effects on the scientific community. MIT cites from an enthusiastic email to the News Office: "In the long run, it will speed up the progress of mankind" [MIT TECH TALK 2001]. Other sources compare OpenCourseWare with the great libraries of the renaissance, in which all scientific knowledge of that time was stored and made available and call it the "Napster for Scientists" [RADEMACHER 2001].

4 A CRITICAL COMPARISON

The following part compares the open source software and the open courseware approach. It takes a look at specific properties of MIT's OpenCourseWare as well as a more generalized concept of open courseware. The basic nature of a product as well as the motivation of contributors on an individual

and institutional level influence the suitability of a product or service for an open approach. Existing products or services in the public domain on the other hand have some influence on the nature of the product (e.g. by providing a publicly available basis for further development) as well as on the motivation (e.g. by enhancing the visibility of the contributors). Figure 1 illustrates these basic relationships.

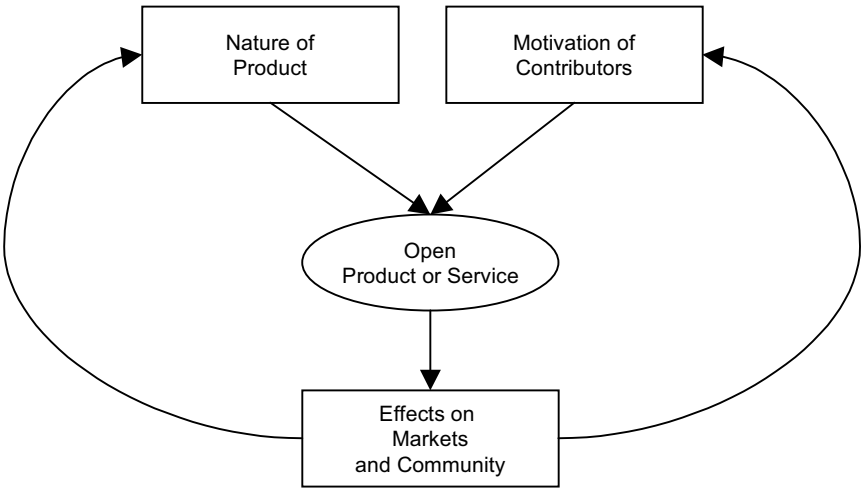

Figure 1: Overview of causal effects

4.1 Comparison in Nature

Open source software and open courseware both contain codified, explicit knowledge. Open source software is a computer program for a specific purpose and contains coding knowledge at the same time. Following the hierarchical view of data, information and knowledge [ALAVI and LEIDNER 2001] the executable program is just data to the user – as he does not understand the internal mechanisms. The advanced programmer on the other hand receives information in the form of source code and is able to transform it into knowledge in his mind, as a large shared knowledge space exists. Computers with standard operating systems, standard programming languages and an online community create this context.

Open courseware is teaching knowledge both in content and structure. It can be of interest for persons willing to learn about the course topic as well as for teachers who want to see how others are teaching specific subjects or want to integrate it into their own teaching material. Course material is almost always information (i.e. processed data). Depending on their background students and teachers will be able to transform this information into useful knowledge. But the shared context (e.g. the same curriculum, common classroom experience) for online course material is in general smaller than for open source software. From a constructivist model of learning (e.g. [VON GLASERSFELD 1995]) interaction with experienced human instructors is needed to create a shared knowledge space. This is the precondition for transforming the information from the course material into knowledge.

Besides this social context for using a courseware also a common technological basis is missing. While open software is built on a common infrastructure of tools, languages, and development environments – i.e. mainly UNIX based environments and C or JAVA programming languages – no similar widely accepted common framework exists for producing courseware (e.g. Learning Objects [WILEY 2001]). Adobe Acrobat documents and Microsoft PowerPoint are presently the smallest common denominator.

Both products are virtual and they both can be distributed and copied at very low cost resulting in a possible strong positive network effect. They both represent the fruits of research and endeavor and are generally perceived as important and valuable. Contributions do not depend on proprietary techniques but on knowledge that is available to a significant group of people – i.e. software

developers or teachers. Both can be the base for multiple complementing commercial services – offers like teaching or maintenance.

Applying the Open Source Definition one might say that courseware is the source code of university teaching. MIT's OpenCourseWare will be available free of charge in a readable format. Following this reasoning both concepts share the concept of included source code according to article 2 of the Open Source Definition. MIT OpenCourseWare however does not comply with articles 1 and 3 of the Open Source Definition. Article 1 of the Open Source Definition (free redistribution) requires the license not to restrict any party from selling the software. MIT excludes the commercial use of its courseware. Open Source Definition article 3 requires that modifications and derived works must be allowed and may be distributed under the same license as the original software. MIT excludes the co-development of its courseware through its intellectual property policy.

While the open source movement goes back to the 1980s, OpenCourseWare is a recent phenomenon. This means there is a substantial pool of open source software while open courseware is still in its infancy and there is no substantial basis to build on right now. Open source is a concept, which is shared and used by programmers around the world; OpenCourseWare is for now restricted to MIT and its courseware editors. While the development of open source software is often a self-organized and distributed effort for a single product [WEBER 2000] the OpenCourseWare project is centrally coordinated and funded by MIT. There is no established culture of competing solutions and distributed conflict resolution. Furthermore the open source movement also departs from the traditional producer-consumer relationship in software development by involving the software user in the development process, thus creating a new governance structure [LUTTERBECK 2001]. The OpenCourseWare concept at this time does not directly address the involvement of students – although feedback is delivered on the level of individual courses.

A specific problem of courseware is that it typically uses papers and exhibits from copyrighted material. While these can be authorized to be used in a specific course, they normally cannot be made freely available on the Internet [JENSEN 2001]. From a systematic point of view a bootstrapping process needs to be established first to generate complete open course material. Open source software on the other hand has already established numerous building blocks on different levels.

Although both courseware and program source code are complex products, their type of complexity differs. The complexity of software like operating systems is given by the multitude of tasks and the multiple different hardware environments it has to work with. Courseware is to support complex human learning processes. It is applicable in a specific context (e.g. at a certain stage of the MIT curriculum). One course is often developed by a single faculty member and there is no need for flexible modularization. Modularity on the other hand is a basic requirement for distributed development as well as flexible re-use.

Similar to software [WEBER 2000] there is no precise metrics for quality in courseware. The judgment of peers is an important approach in theses situations. While this is an explicit part of the open source culture it is not common in the development of educational content in the big scale and MIT does not invite to a collaborative improvement process at this stage of the OpenCourseWare project. While there are infinite ways to code a program so that it fulfills a specific purpose, whether or not it fulfills its purpose is an objective matter. Even the subjective part of coding, decisions about specific implementation issues, can to some degree be compared objectively in terms of reductions in file size or execution time [OPENCONTENT.ORG 2001]. While there are some comparatively objective aspects courseware, such as accuracy of factual information, the style of presentation and the use of good examples or case studies is often a much more subjective matter.

4.2 Comparison in Motives

A motive, which both concepts have in common, is the political/ethical one of the spirit to share knowledge with other people. The underlying principle is that shared knowledge is often regarded as

more beneficial to society than proprietary secrecy. In both cases there might be a certain pride to show what has been achieved as well as an idealism to give to those, who cannot afford to buy knowledge.

Peer recognition is an important individual motivation for software developers as well as for researchers and teachers. In addition to instrumental (i.e. monetizable) aspects contributors value status and reputation at least in part as symbolic rewards. It has been shown that open source software development has been particularly successful in applications that are most useful for the developers themselves, e.g. operating systems [JOHNSON 2001]. While some open source software exists for word processing and spreadsheets it is still not comparable in quality with commercial office solutions. Thus it can be expected that teaching scientists have an interest to produce high-quality courseware. This is especially true as contributors can gain knowledge themselves by contributing to the project.

The methodological motives of the two concepts differ very much. The central aim of OpenCourseWare is not the improvement of its quality by international cooperation. One might even conclude the opposite: OpenCourseWare aims to show, how well made centrally developed MIT courseware can be.

The economic motives differ as well: For MIT the economic gain is presently much more limited than the gain to the individual programmer developing open source software. The programmer can already rely on substantial efforts of others, which he can use if he accepts the open source license. For MIT it is pretty much the opposite. They did not need to open their courseware in order to make use of (its own) achievements of the past. Whether the gain of reputation of MIT through the OpenCourseWare program and the organizational benefits will compensate for the expected cost of about US$ 100 million is difficult to judge. It is fair to say that MIT has already an excellent reputation, even without OpenCourseWare.

4.3 Comparison of Effects on Markets and Scientific Community

Both concepts are valuable to the scientific community. Open source is a very substantial contribution to the communication between developers and OpenCourseWare can be of great value in improving the teaching of scientific findings to students all over the world. The open access to source code and course material fosters the exchange between people and innovational power [METIU and KOGUT 2001]. On the other hand, the use of intellectual property policy to stimulate inventions needs to balance concerns over free riding with considerations of self-selection [BESSEN 2001].

Scientific progress is hard to measure and compare. The direct effect of OpenCourseWare might be much broader than that of open source, because courseware will be available in many different areas of teaching. The direct benefit of open source to computer science, however, is up to present very high. As OpenCourseWare does not yet exist, the value of any estimates leading to comparisons is very limited. It has been observed however, that good open source projects are developed initially by a small group [JOHNSON 2001]. The MIT project might have the potential to be the foundation of a broader development in this area.

While both concepts challenge the currently dominating paradigm, which advocates for the commercial enterprise model as the best production model, the true long-term impact of the two movements is still open. Even open source software may be a way of organizing information system development that is effective only for certain specialized products [BJÖRK 2001].

5 CONCLUSIONS

There are some similarities and numerous differences between open source software and MIT OpenCourseWare. Most of the similarities can be found in the motives and the effects on the scientific community: Both carry the motive to share knowledge with others worldwide and both can be considered to be beneficial to scientific progress. When comparing the nature of the two approaches

some important differences become apparent: Open source is much more than knowledge sharing. It is a system of software development, a legal framework and a philosophy. OpenCourseWare falls short of most of these elements. It does not include the participation of non-MIT staff in the courseware development. Furthermore it is not an integrative legal framework under which other universities might publish their courseware as well, or a philosophy in itself. Although the project's title invites comparisons with open source software, the analogy is probably limited to the concept of making information freely available, hoping for reciprocal gestures from other institutions [DANIEL and COX 2001].

With respect to the nature of the task as well as to the motivation of the agents courseware incorporates many of the factors that have been identified as supportive for the development of open source software [WEBER 2000]. This is why the idea of open courseware deserves a much larger discussion in academic circles beyond MIT's OpenCourseWare project because of its possible impact on university teaching. A closer look at the open source movement can help to develop the idea and to build an integrative legal framework for sharing and improving courseware. Some models of content sharing come close to open source: Examples are the GNU Free Documentation License [FREE SOFTWARE FOUNDATION 2000], and the OpenContent License [OPENCONTENT.ORG 1998]. These aspects of open development environments for educational knowledge, legal frameworks and philosophy should be considered as well as more technical approaches to make course content interoperable. While the discussion about free access to scientific research papers has received some public coverage (e.g. in [NATURE 2001, SCIENCE 2001]) the development of educational resources deserves a similar debate.

REFERENCES

ALAVI, M. and LEIDNER, D. E. (2001) *Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues.* MIS Quarterly, Vol. 25 (2), 107-136.

BESSEN, J. (2001) *Open Source Software: Free Provision of Complex Public Goods*. Working Paper, Research on Innovation.

BJÖRK, B.-C. (2001) *Open Source, Open Science, OpenCourseWare*. Proceedings *ECAADE - Education for Computer Aided Architectural Design in Europe*, Helsinki, Finland, August 29 - 31, 2001, 13-17, Online: http://www.hut.fi/events/ecaade/E2001presentations/01_01_bjork.pdf (accessed: November 23, 2001).

DANIEL, G. AND COX, K. (2001) *MIT OpenCourseWare - Soon Free on Your PC.* Web Tools Newsletter, Online: http://webtools.cityu.edu.hk/news/newslett/mit.htm.

DIBONA, C., OCKMAN, S. and STONE, M. (1999) *Open Sources: Voices from the Open Source Revolution.* O'Reilly & Associates, Sebastopol, CA.

FREE SOFTWARE FOUNDATION (1991) *GNU General Public License*. Online: http://www.gnu.org/copyleft/gpl.html (accessed: November 19, 2001).

FREE SOFTWARE FOUNDATION (2000) *GNU Free Documentation License*. Online: http://www.gnu.org/copyleft/fdl.html (accessed: November 20, 2001).

HIMANEN, P. (2001) *The Hacker Ethic and the Spirit of the Information Age.* Random House, New York.

HIPPEL, E. V. (2001) *Learning from Open-Source Software.* MIT Sloan Management Review (Summer 2001), 82-86.

JENSEN, B. (2001) *Threads Regarding Professors Who Share*. Online: http://www.trinity.edu/rjensen/openshare.htm (accessed: November 21, 2001).

JOHNSON, J. P. (2001) *Economics of Open Source Software*. Working Paper, Cornell University.

LUTTERBECK, B. (2001) *Unexploited Resources of Online Education for Democracy - Why the Future Should Belong to "OpenCourseWare"*. Proceedings *Innovations for an e-Society. Challenges for Technology Assessment*, Department of Computer Science, Technical University Berlin, Berlin, October 17-20, 2001, Online: http://ig.cs.tu-berlin.de/bl/063/Lutterbeck2001-ELearning.pdf (accessed: November 24, 2001).

METIU, A. and KOGUT, B. (2001) *Distributed Knowledge and the Global Organization of Software Development*. Working Paper, The Wharton School, University of Pennsylvania, Philadelphia, February 2001, Online: http://knowledge.wharton.upenn.edu/PDFs/1003.pdf (accessed: November 23, 2001).

MIT (2001a) *MIT OpenCourseWare*. Online: http://web.mit.edu/ocw/ (accessed: October 25, 2001).

MIT (2001b) *MIT OpenCourseWare - Faculty Views*. Online: http://web.mit.edu/newsoffice/nr/2001/ocw-fac.html (accessed: November 22, 2001).

MIT (2001c) *MIT OpenCourseWare Fact Sheet*. Online: http://web.mit.edu/ocw/ocw-facts.html (accessed: October 25, 2001).

MIT TECH TALK (2001) *Plan evokes worldwide response*. Online: http://web.mit.edu/newsoffice/tt/2001/apr11/ocwside.html (accessed: October 25, 2001).

NATURE (2001) *Nature Webdebates, Future e-access to the primary literature*. Online: http://www.nature.com/nature/debates/e-access/ (accessed: November 20, 2001).

OPEN SOURCE INITIATIVE (2001) *The Open Source Definition - Version 1.9*. Online: http://www.opensource.org/docs/definition.html (accessed: October 25, 2001).

OPENCONTENT.ORG (1998) *OpenContent License (OPL)*. Online: http://opencontent.org/opl.shtml (accessed: November 20, 2001).

OPENCONTENT.ORG (2001) *Open Source Content Development*. Online: http://opencontent.org/bazaar.shtml (accessed: November 20, 2001).

RADEMACHER, H. (2001) *Napster für Naturwissenschaftler.* Frankfurter Allgemeine Zeitung (85), 51.

RICHARDS, P. (2001) *MIT to make nearly all course materials available free on the World Wide Web*. Online: http://web.mit.edu/newsoffice/nr/2001/ocw.html (accessed: October 25, 2001).

SCIENCE (2001) *Science Discussion - A Public Archive for the Primary Scientific Literature*. Online: http://www.sciencemag.org/feature/data/hottopics/plsdebate.shtml (accessed: November 20, 2001).

VON GLASERSFELD, E. (1995): *Sensory experience, abstraction, and teaching*. In: Steffe, L. and Gale, J. (eds.) *Constructivism in education*, Lawrence Erlbaum Associates, Inc., New Jersey, 369-384.

WAYNER, P. (2000) *Free for All - How Linux and the Free Software Movement Undercut the High-Tech Titans.* Harper Business, New York.

WEBER, S. (2000) *The Political Economy of Open Source Software*. BRIE Working Paper, No. 140, University of Berkeley, June 2000.

WIGGINS, R. W. (2001) *MIT Launches OpenCourseWare Initiative on the Web.* Information Today, Vol. 18 (6), 22-24.

WILEY, D. A. (ed.) (2001) *The Instructional Use of Learning Objects*. Agency for Instructional Technology.