Department of Computer Science,
University of Otago

Technical Report OUCS-2004-14

Open Source Intellectual Property Rights - Issues and Trends

Author:
Stewart Fleming
Department of Computer Science, University of Otago

Status: to appear as book chapter in: *Encyclopedia of Multimedia and Technology*
(M Pagani ed). IDEA Group Publishing, 2005

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/research/techreports.html

DRAFT ENCYCLOPEDIA SUBMISSION

Open Source Intellectual Property Rights

Issues and Trends

Stewart T. Fleming
Department of Computer Science,
University of Otago, New Zealand

INTRODUCTION

The *Open Source Software movement* exists as a loose collection of individuals, organizations and philosophies roughly grouped under the intent of making software source code as widely available as possible (Raymond, 1998). While the movement as such can trace its roots back over 30 years to the development of academic software, the Internet and World Wide Web etc, the popularization of the movement grew significantly from the mid-80's (Naughton, 2000).

The *Free Software movement* takes open source one step further, asserting that, in addition to freedom of availability through publication, there should be legally-enforceable rights to ensure that it stays freely available and that such protections should extend to derived works (Stallman, 2002).

The impetus of both movements has resulted in the widespread distribution of a significant amount of free software, particularly GNU/Linux and Apache Web server. The nature of this software and the scale of installation appears to be an emerging concern for "closed" software vendors. At this time, we are seeing the emergence of legal challenges to the open source movement and a clash with the changing landscape of intellectual property and copyright protection.

There is spirited debate within and between both movements regarding the nature of open source software and the concerns over the extent to which software should remain free or become proprietary. This article concentrates on the issues directly relating to open source licenses and their impact on copyright and intellectual property rights and the legal risks that may arise. For more general reference, the reader is directed to the Web sites of the Free Software Foundation (http://www.fsf.org), Open Source Initiative (http://www.opensource.org) and the excellent bibliography maintained by Stefan Koch (http://wwwai.wu-wien.ac.at/~koch/forschung/sw-eng/oss_list.html)

BACKGROUND

Motivations for Participation

The open source software movement is motivated by the desire to make software widely available to stimulate creative activity (either in the development of derivative software, or in the use of that software in other endeavours). Free software requires open source software and goes further by pursuing the protection of ideas – ensuring that the intellectual basis for a software development can never be exclusively controlled or exploited.

Why would an individual decide to participate in a movement where they might not accrue any direct financial benefit? Boyle (2003) discusses individual motivations with the notion of a “reserve price” – a level at which any individual decides for themselves to become an active participant rather than a consumer and to engage in some voluntary activity. It might be out of altruistic motives; it might be for the intellectual challenge; it might be to solve a personal problem by making use of collaborative resources (and the entry level to collaboration is participation).

Gacek & Arief (2004) identify two additional motivations for open source participation: “developers are users” (p35) and “knowledge shown through contributions increases the contributor’s merit, which in turn leads to power.” (p37). This indicates a powerful motivation through self-interest and enhanced reputation with wide recognition of contributions, especially in large projects.

The presence of large industrial consortia in the open source movement and broad participation across many software development companies indicates that many commercial organizations are also motivated to participate. Table 1 lists broad categories to explain individual, academic institution and commercial motivations to participate in open source production activity.

Table 1: Motivational factors governing open source participation.

Individual

- Intellectual challenge.
- Desire to improve the greater good.
- Expectation of improved employment prospects.
- Desire for fame and associated publicity.

Academic Institution

- Costs accounted for.
- Expectation of greater prestige.
- Encouragement of staff development.
- Outlet for intellectual capital of the organization.

Commercial Enterprise

- Excess of R&D expertise/capability.
- Reduction in operational cost by benefiting from collaborative activity e.g. problem-solving, ideas generation etc.
- Outlet for intellectual capital of the organization.

Dempsey *et al* (2002) liken participation in open source software development to peer review in scientific research. By releasing one’s software and using the software of others, continual innovations and improvements are made.

The large-scale collaborative nature of open source software development makes it important that the contributions of individuals are recognized and the resulting situation is that ownership of any piece of open source software is jointly held. The solution that has evolved in the development of free software and open source movements has been the development of a variety of licensing models to ensure recognition and availability of contributions.

Open Source Definition

The Open Source Initiative (OSI) was begun in 1998 to make the case for open source software development more accessible to the commercial world. It provides samples of open source licenses and ratifies many of the licenses that cover various open source software developments. The Open Source Definition (Perens, 1999) is a useful description of the characteristics of what constitutes open source software (Table 2)

Table 2: OSI Definition of open source software.

1. Free distribution – no royalty can be charged for the distribution of software.
2. Source code – software must be distributed in source code form.
3. Modifications and derived works – the license must allow a licensee to make changes or create derived works based on the original and allow them to be distributed under the terms of the original license.
4. Integrity of the author's source code – the license can specify that the original code must be distributed and then "patched" to incorporate modifications, thus preserving the integrity of original work.
5. No discrimination against persons or groups.
6. No discrimination against fields of endeavour.
7. Distribution of license – no additional licensing terms can be required.
8. License must not be specific to a particular software distribution.
9. License must not restrict other software.
10. License must be technology-neutral – provision of the license cannot depend on explicit assent or acceptance from the user.

See also: http://www.opensource.org/docs/definition.php

Software licenses that meet this definition can be considered as open source licenses and the OSI provides certification for conforming licenses. While there have been many open source and free software licenses that have been created to suit various purposes, there are three main influences that will be considered in this article: GNU General Public License models, BSD license models and Mozilla Public License (MPL) models.

MAIN FOCUS: OPEN SOURCE LICENSE MODELS

One of the most important developments to come out of the open source movement has been the proliferation and deep consideration of various licensing models to grant various rights to users of software. The collaborative depth of the movement is neatly illustrated by the spirited debate that surrounds issues that affect the community as a whole and the diversity of the community provides broad viewpoints that cover all aspects, from the deeply technical to the legal. There is a wide range of different licenses (http://www.fsf.org/licenses/license-list.html) , some free software, some not-free and others incompatible with the General Public License (GPL). Table 2 summarizes the restrictions on various development activities applied by the three common classes of license.

Table 3: Classification of restrictions imposed by various licenses.

License	OSI Compatible	GPL Compatible	Redistribution	Charging	Copyleft
GPL	✓	✓	Licensee must guarantee availability of entire code	Licensee may not charge for the code or royalties, but may charge for distribution cost	Derived work is subject to GPL
LGPL	✓	✓	As above	As above	×
BSD	✓	✓	No restriction	No restriction	×
MIT	✓	✓	No restriction	No restriction	×
MPL	✓	✓	Licensee must guarantee availability of source code for at least 12 months after first availability	Royalty-free	✓

BSD

The modified Berkeley Systems Development (BSD) license is an Open Source license with few restrictions and no impact on derived work. It requires only that attribution of copyright be made in source code and binary distribution of software. It specifically excludes any software warranties and disallows the use of the original organization in any advertising or promotion of derived works.

MIT(X11)

This is another open source license with very few restrictions and no impact on derived works. It requires only that a copyright notice be included with copies or substantial extracts of the software and excludes warranties.

The risk with unrestricted licenses such as BSD and MIT models above is that a licensee can produce a derived work and not release improvements or enhancements, which might be useful to the wider community (Behlendorf, 1999).

Mozilla Public License

The modified version of the Mozilla Public License (MPL) (http://www.opensource.org/licenses/mozilla1.1.php) is a free software license that meets the OSI definition and is compatible with the GPL. It contains a number of complex provisions, but the inclusion of a multiple licensing clause allows it to be considered compatible with the GPL. The license is the controlling license for the Netscape Mozilla Web browser and associated software. It was developed specifically for the business situation at Netscape at the time of release, but has since been used in many open source developments. The MPL/GPL/LGPL tri-license (http://www.mozilla.org/MPL/boilerplate-1.1/mpl-tri-license-txt) provides the mechanism for maintaining compatibility with the GPL.

The license includes clauses that are intended to deal with the software patent issue where source code that infringes on a software patent is deliberately or inadvertently introduced into a project. Behlendorf (1999) points out that there is a flaw in the waiver of patent rights in the license, but suggests that in general, the license is strong enough to support end-user development.

GNU General Public License

The GNU General Public Licenses (GNU GPL or GPL for short) was originally developed by Richard Stallman in around 1985 with the specific intention of protecting the ideas underlying the development of a particular piece of software. Free software does not mean that software must be made available without charge; it means that software, once released, must be always freely available. The GPL is a free software license that incorporates the 'copyleft' provision that makes this freedom possible.

"To copyleft a program, we first state that it is copyrighted; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code or any program derived from it but only if the distribution terms are unchanged. Thus, the code and the freedoms become legally inseparable." (http://www.gnu.org/copyleft/copyleft.html)

The GPL has provoked much debate and the deliberate inclusion of political overtones in the wording of the license make it unpalatable to some. Indeed, the Lesser GNU Public License (LGPL) is essentially the same as the GPL, but without the copyleft provision. This makes a free software license option available to commercial software developers without the obligation to release all of their source code in derived works.

In March 2003, The SCO Group based in Utah, USA initiated a lawsuit against IBM alleging that proprietary SCO Linux code had been integrated into Linux, the leading open source operating system, and seeking damages since IBM has non-disclosure agreements in place with SCO regarding UNIX source code. The SCO Group has also sent letters to over 1500 large companies advising them that they may face legal liability as Linux customers under the terms of the GPL. It is of great interest that the SCO-IBM lawsuit specifically targets the GPL, which links source code with a legally-protected freedom to distribute and make use of in derived work.

Whatever the motivations behind the lawsuit and its eventual outcome, as part of risk management activity, developers should be aware of the implications of creating and using software that is covered by the various licenses (Välimäki, 2004).

Adopting licenses other than the GPL weakens both it and the overall argument in favor of free software. This may in fact be the intention of the lawsuit, to mount a legal challenge that, if successful, would strongly dissuade developers from using the GPL.

Another barrier to the proliferation of open source software lies in the need to create broad-based standards. Without standards, inter-operation of software created by multiple developers is difficult to achieve. However, the presence of patents that protect particular software inventions raises problems for the adoption of standards by the open source community. If there is only one way to accomplish a certain outcome, and that method is protected, then development of an open source version is effectively blocked. Even if patents are licensed by their owners specifically for use in the development of open standards, there is an incompatibility with the GPL regarding freedom to create derived works (Rosen, 2004).

CRITICAL ISSUES AND FUTURE TRENDS

There are several issues that emerge from the consideration of open source and free software. As open source and free software becomes more widely used in different situations, potential legal risks become greater. Software development organizations must be aware of the possible effects of open source licenses when they undertake open source development.

The wide participation required by large-scale open source software development raises the risk of infringement on intellectual property, copyright, or software patent. The exclusion of warranties for software defects in most open source software licenses should cause organizations considering the adoption of open source software to carefully consider how quality and reliability can be assured.

The World Intellectual Property Organization survey of Intellectual Property on the Internet (WIPO, 2002) identifies Open Source software as the source of emergent copyright issues (p43). It does not give any special treatment to the moral rights of authors with respect to software and such rights are variable across international jurisdictions (Järvinen, 2002). Since the enhancement of reputation is an important motivating factor in participation in open source software development, software authors might benefit from more uniform international recognition of their right to assert authorship and their right to avoid derogatory treatment as author of a work.

Quality and reliability characteristics of open source software raise concerns for organizations in areas where certification is needed such as in mission-critical activities, medicine. Harris (2004) provides an interesting account of how open-source software was incorporated into the mission-critical data analysis tools for the Mars rovers Spirit and Opportunity. Zhao & Elbaum (2003) report that although there was wide user participation in open source software projects that they surveyed, and tools to track software issues were commonly used, the nature of testing activities was often shallow and imprecise. The lack of formal tools for testing, especially test coverage and regression testing should lend a note of caution to those considering the use of open source software. The onus is on software developers making use of open source software to be duly diligent in their testing and integration of software.

A significant potential risk to open source software development is the protection of closed software markets by enforcement of software patents. An organization that has been granted a software patent for some algorithm or implementation is granted the rights to charge royalties for use, or may force others to cease distribution of software that employs the scheme covered by the patent. Open source software is vulnerable to this form of restriction since all source code is publicly-available. On the other hand, the distributed nature of the open source community can be a buffer against this form of restriction (Järvinen, 2002)

If we consider free software, the terms of the GPL have been written with reference to US law. Work is required to validate the terms of the license with respect to other jurisdictions. The main concern with the GPL is the copyleft clause covering derivative works. Järvinen (2002) has considered the GPL with respect to Finnish law; Välimäki (2001) gives a good account of the differences between US and European Union treatment of derivative works. Metzger & Jaeger (2001) have found that although the GPL is generally compatible with German law, there may be issues with the complete exclusion of warranties. This may be the case in other jurisdictions where consumer protection laws are in force (*e.g.* US, EU, Finland, New Zealand) and warranties cannot be excluded. In the US, the lawsuit mounted by SCO *vs.* IBM in March 2003 is seen by many as a direct challenge to the GPL.

The exact nature of derivative works is determined by the courts. Välimäki (2004) summarizes different interpretations for what constitutes a derivative work (Table 4). Many of the issues regarding what does or does not constitute a derivative work are held only by mutual agreement amongst those in the open source software community. Software development organizations must be aware of the implications of open source software licenses, not only to cover the software that they distribute, but also those that cover any software they might use in the development. There is a serious risk of inadvertent breach of the GPL where an organization uses software covered by the GPL in proprietary software that it develops. Until there is a firm legal resolution in favour or against the terms of the GPL, there is no firm basis for the application of the principles underlying the GPL.

Table 4: Interpretation of derivative works (Välimäki, 2001, Ravicher, 2002, Webbink, 2004).

Generic theories

1. Idea-expression dichotomy – ideas cannot be copyrighted, only the expression of ideas. Deriving a work based on study of the original would be permissible, but direct copying of the source text would infringe copyright.

2. Abstraction-filtration-comparison – contextual analysis of source code structure and content to determine overall similarity.

Technical interpretations

1. Component-based – if a software component depends on another for its function, it might be considered as a derivative work. Common acceptance among open source developers is that 'static' linking (where a function is compiled into the executable) constitutes a derived work; "dynamic" linking (an external function call) does not.

2. Device-drivers and plug-ins – under interpretation of the free software movement, if a program shares data structures and is linked dynamically at run-time, it constitutes a derived work.

3. Simple *use* of a program is not considered to be a derived work.

In more general terms, the exact nature of security and liability with regard to open software is hard to establish. Kamp (2004) provides an interesting anecdote about the unimagined scale of distribution of a single piece of open source software. Although one of the much-vaunted strengths of the open source community is that "many eyes make all bugs shallow" (Raymond, 2001, p30), security issues may still be difficult to identify and resolve (Payne, 2002). Peer-review of public software is an advantage, but successful outcomes still depend on the motivation of properly-skilled individuals to methodically study, probe and fix open source software problems.

CONCLUSIONS

The future for open source licenses will be determined by the outcomes of legal challenges mounted in the coming years. The interpretation of many aspects of the GPL can only be clarified properly through the courts of law. The interpretation in various jurisdictions will affect the international applicability of such licenses. Such tests are to be welcomed – they either confirm the strength of the open source and free software movements or through a competitive influence, they cause them to reorganize in order to become stronger.

REFERENCES

Behlendorf, B. (1999). Open Source as a Business Strategy. In C. DiBona, S. Ockman & M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly & Associates.

Boyle, J. (2003). The Second Enclosure Movement and the Construction of the Public Domain. *Law and Contemporary Problems, 66*(33), 33-74.

Dempsey, B. J., Weiss, D., Jones, P., & Greenberg, J. (2002). Who Is An Open Source Software Developer? *Communications of the ACM, 45*(2), 67-72.

Gacek, C., & Arief, B. (2004). The Many Meanings of Open Source. *IEEE Software, 21*(1), 34-40.

Harris, J. S. (2004). Mission-Critical Development with Open Source Software: Lessons Learned. *IEEE Software, 21*(1), 42-49.

Järvinen, H. (2002). *Legal Aspects of Open Source Licensing*. Helsinki, Finland: University of Helsinki, Department of Computer Science.

Kamp, P.-H. (2004). Keep in Touch! *IEEE Software, 21*(1), 45-46.

Metzger, A., & Jaeger, T. (2001). Open Source Software and German copyright law. *International Review of Industrial Property and Copyright Law, 32*(1), 52-74.

Naughton, J. (2000). *A Brief History of the Future*: Phoenix Press.

Payne, C. (2002). On the security of open source software. *Information Systems Journal, 12*(1), 61-78.

Perens, B. (1999). The Open Source Definition. In C. DiBona, S. Ockman & M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly & Associates.

Ravicher, D. (2002). *Software Derivative Work: A Circuit Dependent Determination*. New York: Patterson, Belknap, Webb and Tyler.

Raymond, E. S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly and Associates.

Rosen, L. (2004). *Open Source Licensing, Software Freedom and Intellectual Property Law*. New York: Prentice Hall.

Stallman, R. (2002). *Free Software, Free Society: Selected Essays of Richard M Stallman*: GNU Press.

Välimäki, M. (2001). *GNU General Public License and the Distribution of Derivative Works.* Paper presented at the Chaos Communication Congress, Berlin, Germany.

Välimäki:, M. (2004). A Practical Approach to the Problem of Open Source and Software Patents. *European Intellectual Property Review, (forthcoming)*.

Webbink, M. H. (2004). *Open Source Software.* Paper presented at the 19th Annual Intellectual Property Conference, Washington, DC.

World Intellectual Property Organization. (2002). *Intellectual Property on the Internet: A Survey of Issues* (No. WIPO/INT/02). Geneva, Switzerland.

Zhao, L., & Elbaum, S. (2003). Quality assurance under the open source development model. *Journal of Systems and Software, 66*, 65-75.

Terms and Definitions

Open source software: computer software distributed under some licence that permits the user to use, modify (including the creation of derived works) and distribute the software and any derived work, free from royalties.

Free software: software that is distributed under the terms of a licence agreement that make it freely available in source code form. Strong advocates of free software insist that the ideas underlying a piece of software, once published, must always be freely available.

Copyright: protected right in many jurisdictions that controls ownership over any material of a creative nature originated by an individual or organization.

Copyleft: provision in the GNU General Public License that forces any derived work based on software covered by the GPL to itself be covered by the GPL. That is, the author of a derived work must make all source code available and comply with the terms of the GPL.

Intellectual property (IP): wider right to control ownership over any material of a conceptual nature – invention, idea, concept – as well as encompassing material originally covered by copyright.

Ownership: the association of the rights over intellectual property either with an institution or an individual so as to enable exploitation of that IP.

1. **Work for hire** – an individual employed by an institution produces materials that are owned by the institution.
2. **Assertion of copyright** – retention of the protection right of copyright by an individual and hence the ability to collect any royalties that may be apportioned.
3. **Ownership by contract** – transfer or otherwise licensing all or part of copyright from the owner to one or more other parties covered by an explicit contract.

Open source licensing model: a statement of the rights granted by the owner of some piece of open source software to the user.

1. General Public License – specifically links source code to legally-protected freedom to publish, distribute and make use of in derived works.
2. Attribution – source code published under this license may be used freely provided that the original author is attributed.
3. Standard setting – source code under such a license may be used only in activity that defines an industry standard.
4. Shareware – software is available to users only on payment of a nominal fee.

Licensing domain: characterization of the breadth of availability and level of access to open materials.

1. Public domain – owned by the “public at large”.
2. Publicly available – obtainable for free or minimum cost of materials.
3. Freely available – wide distribution at no cost to consumer.
4. Open source (see **open source licensing model** for strategies).