A Practical Approach to the Problem of Open Source and Software Patents

Mikko Välimäki LL.M., University Lecturer, Helsinki University of Technology mikko.valimaki@hut.fi, +358-50-5980498, http://www.valimaki.org/

1. Introduction

The role of open source and free software development has become central in many infrastructural software projects. A great example is the Internet, which has practically grown on open source software. At the moment, popular open source products such as GNU/Linux operating system, Apache web server and several databases are also used increasingly within corporate networks. One could argue that open source forms perhaps the fastest growing and most innovative sector of the software industry today.

However, many individual software developers and small companies have warned that software patents threat the functionality of the open source model.¹ Recent political campaigning around the EU's software patent directive proposal has increased fears that distributed open development is particularly vulnerable to patent monopolies.² Well-known open source developers including Linus Torvalds have expressed critical public opinions on software patents in general and the directive proposal in particular.³ The critique has been mainly centered on economic arguments: extensive software patents may threat software innovation.⁴

¹ Warnings have a long history in developer communities: there has been campaigning against software patentability since the early 1990s. See e.g. The League for Programming Freedom: Against Software Patents, February 28, 1991,

http://lpf.ai.mit.edu/Patents/against-software-patents.html

² See Proposal for a directive of the European Parliament and of the Council on the patentability of computer-implemented inventions, Brussels, 20.02.2002, COM(2002) 92 final. Most active critique has been arranged by Foundation for a Free Information Infrastructure, a group run by Hartmut Pilch. See http://www.ffii.org/.

³ For example Linux developers Linus Torvalds and Alan Cox sent an open letter to the European Parliament in the fall 2003 expressing their deep concerns. See http://www.effi.org/patentit/patents_torvalds_cox.txt

⁴ Political debate at EU level on software innovation is not new either. Lobbying on the Council Directive of 14 may 1991 on the Legal Protection of Computer Programs, 91/250/EEC, centered on the questions of whether reverse engineering and copyright over interfaces would threat software development and innovation. See Band, Jonathan and Katoh, Masanobu: Interfaces on Trial: Intellectual Property and Interoperability in

In this article we review several suggestions to solve the patenting problem with open source development. These include patent pools for open source developers, aligning development more closely with the patenting process, solving the problem of trivial patents and introducing new liability exceptions to the patent law.

We conclude the paper by arguing that in the case patents and open source development have conflicts, the legal system should be improved in the first place. It would not be desirable to require open source development as a methodology to adopt abstract rules of law that may not be in par with the software development reality. Therefore the liability exception policy might offer the best outcome for all.

2. Open Source Licenses and Patents

Let's start from looking at the patenting problem from open source licensing perspective. Software projects can be identified as open source when the software is licensed with an open source license.⁵ Open source licenses can be further categorized into several groups.⁶ From patents perspective, the most important licenses are GNU General Public License (GPL) and GNU Lesser General Public License (LGPL), which are also the most used open source licenses. Both of these licenses include a similar patent clause.⁷ GPL clause 7 reads as follows:

"7. If, as a consequence [...] of patent infringement [...] conditions are imposed on you [...] that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

⁵ For a complete list of open source licenses, see http://www.opensource.org/licenses/

the Global Software Industry, Westview Press, 1995. What is new in the patenting debate, however, is the strong role of individual developers and activists.

⁶ Metzger, Axel – Jaeger, Till: Open Source Software, Verlag C.H.Beck, 2002. See also http://www.ifross.de/ifross_html/lizenzcenter.html

⁷ The patent language in GNU licenses dates back to 1991, when the license author Richard M. Stallman and the Free Software Foundation already campaigned against software patents in the United States.

license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. [...]"

The preamble of GPL explains the motivation behind this obligation:

"[...] any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all."

In short, GNU GPL and LGPL have a built-in termination mechanism that does not allow the development of software that requires any kind of license payments for third party patents. In more technical wording, GPL and LGPL are incompatible with patent licensing fees: if there is a patent for some software invention and that patent is not licensed for free to every (L)GPL user forever, it is not possible to develop free software for that invention.

Not all open source licenses have such patent clauses. For instance the popular BSD license lacks one. However, patent clauses essentially similar to those of GPL and LGPL are becoming more and more common. Almost all recent open source licenses have such.⁸ Whatever one may think of the practicality of this kind of termination clauses, it seems clear that open source development becomes problematic indeed if there are many existing software patents around.

3. Open Source Development Process

From development perspective, open source and free software are ambiguous concepts.⁹ There are both commercial and non-

⁸ Some of them may have even more far-reaching termination clauses that try to affect the enforceability of other software patents as well. See e.g. the latest versions of Apache license and Open Software Licenses for that matter.

⁹ In this paper, the term "open source" covers also free software. Differences between the meaning of these two terms is mainly philosophical and political. See Stallman, Richard

commercial open source development projects. Some projects are coordinated by a legal entity, some by informal groups of developers.¹⁰

Common to all open source development is the transparency and incremental nature of all development. This means that:

- All source code including potential software inventions is public and commented from the beginning; there are no periods of secrecy and neither is patent data searched before publishing
- Development is distributed in a sense that the number of contributors is in principle unlimited and their identity may be anonymous
- Development is incremental meaning that typically contributions only cover a specific part or function of the program. A project may be abandoned by the initial developers and later continued by others.

There may be both individuals and organizations participating in the development process. The hierarchy and organization of the development process may not be visible to outside. A typical open source project does not require high resources to be technically efficient. Also, a successful project may gain high visibility and large user base without significant commercial marketing efforts.

From the short description above, we can find numerous reasons why the patent system does not fit open source development process at all.¹¹ It should be stressed that open source development has so far worked fine and produced new innovations without anyone applying or licensing patents. It is not in the scope of this paper to study whether the patent system works efficiently with software development in general. It should be clear, however, that the potential setbacks to open source software development caused by an inefficient and malfunctioning patent system are far greater than they would be to other areas of innovative activity.

- M.: "Why "Free Software" is better than "Open Source"",
- http://www.gnu.org/philosophy/free-software-for-freedom.html
- ¹⁰ A good overview of open source development process is e.g. Feller, Joseph –

Fitzgerald, Brian: Understanding Open Source Software Development, Addison-Wesley, 2002.

¹¹ These are discussed in more detail below in section 5.

4. Risk of Patent Infringement with Open Source

The biggest challenge with software patents is that the infringement risk is difficult to measure and manage. It is true that statistically patent infringement cases are rare, many patents are held invalid and that a mere infringement claim does not yet trigger the termination clause for instance in GNU GPL.¹² In practice, however, an open source project faced with a patent infringement claim from a credible corporation may have to terminate just because it would be too costly and time-consuming to find out what the real risk is. In addition to developers also open source users face the infringement risk. Open source licenses do not help since they typically spread the risk from any intellectual property rights infringement to the user.

Most troublesome are those software patents that are not known *ex ante*. Software users may be liable for infringing such patents whether the infringement has been intentional or not. In practice, only the infringement of well-known software patents can be avoided to some extent. These include for example popular audio and video for mat patents (such as MP3, DVD and DivX encoding and playback). Also it is known that Microsoft's widely used file systems and file formats are covered by a number of patents.¹³ Open Source Risk Management Inc., who sells patent insurances, claims that Linux kernel could infringe almost 300 different patents.¹⁴

So what are the practical options to react when a patent infringement claim arrives? First, the use of the invention can be stopped. Second,

 ¹⁴ See Open Source Risk Management's press release: "Results of First-Ever Linux Patent Review Announced", 2nd August 2004. Available at

¹² For example Open Source Initiative's legal counsel Lawrence E. Rosen has stressed that because of these reasons open source developers should not be too worried about patents. See his article "Patents in an open source world", NewsForge, July 27, 2004. Interestingly, a leaked memo from Hewlett-Packard shows how real many companies see potential patent risks with open source. See "HP memo forecasts MS patent attacks on free software", NewsFroge, July 19, 2004.

¹³ For MP3 patent licensing, see http://www.mp3licensing.com/, for DVD licensing, see http://www.licensing.philips.com/licensees/conditions/dvd/, for DivX licensing, see http://www.divx.com/divx/licensing/. At the moment Microsoft licenses FAT filesystem and it is possible that in the future also the widely used NTFS will become under a similar patent licensing program. Also, their new XML-based file formats are under patent licenses. For Microsoft's patent licensing, see http://www.microsoft.com/mscorp/ip/

http://www.osriskmanagement.com/

the patent can be analyzed and determined whether either a license should be negotiated or a new implementation around the patent written. While legally perhaps the safest option, writing a new implementation takes resources and some extra motivation might be needed to "invent the wheel" again. Licensing has its problems too: the patent license should be practically free-of-charge because of the terms of open source licenses (such as GPL clause 7 noted above) and because most individual developers and non-commercial projects couldn't afford any fees in the first place. Now how many patentowners would be ready for that? Finally, it is difficult to argue why anyone should buy a license to an invention, which he cannot in many cases even utilize (there is no source code in patents).

The most recent option to minimize patent infringement risk is insurance. Companies who actually own the copyright and patents in their (open source) software may naturally sell warranties and indemnification to their users.¹⁵ A "pure" open source software product, which no one owns and controls, is a more complicated animal to insure. However, some big IT and insurance companies have started to offer separate warranties and insurances for both the developers and users of open source software including Linux.¹⁶ A liability insurance may be therefore a viable risk management option for instance to small companies who use open source extensively in their business.

5. Policy Debate on Open Source and Software Innovation

Critical policy debate on software patents has been active for years.¹⁷ The opponents of software patents – mainly individual developers, activists with various backgrounds, and small companies – have gained high visibility with comparably lower resources to propatenting advocates. The debate is currently most relevant in the EU

¹⁵ This business model is called dual licensing. For more details see e.g. Välimäki, Mikko: "Dual Licensing in Open Source Software Industry", Systemes d'Information et Management, Vol. 8, No. 1, pp. 63-75, 2003. Available at http://www.valimaki.org/

¹⁶ See e.g. different Linux indemnification policies introduced by HP, Novell, and Red Hat. Also some insurance companies have introduced "IPR Insurance" policies to comprehend traditional corporate liability insurances. See note 14 above.

¹⁷ See footnote 1 and e.g. Nichols, Kenneth: Inventing Software, Quorum Books, 1998, p. 103- and http://www.bustpatents.com/software.htm

where several consultation and research reports have been published on the issue.¹⁸

From open source perspective the most important problems have been said to be:

- Because of open source code patent infringements are relatively easy to detect and prove
- Low resources do not allow patent search and legal defense in the event of infringement claims based on trivial patents
- Especially free software ethics and philosophy are strongly against the use of patents of any kind in software development
- Source publication on the Internet may be interpreted to happen in all jurisdictions and hence infringe potential patents anywhere¹⁹
- Non-open source software developers may compete against open source developers with patent infringement claims; if their patents are held valid and open source developers would require to obtain licenses, there is no guarantee on the terms of these private license agreements between open source development community and commercial software company

Suggested solutions in the recent studies and reports have varied from:

- Solving the problem of trivial patents (Bekkels)
- Founding a patent pool for the open source community (Pbt Consultants)
- Adjusting the development model to include patentable research (Nichols)
- Special liability exceptions to open source developers (Blind et al)

¹⁸ See e.g, Bakels, Reinier: The Patentability of Computer Programmes, European Parliament Directore-General for Research Working Paper, 2002; Blind, Knut – Edler Jakob – Nack, Ralph – Strauss, Joseph: Mikro und makroökonomische Implikationer der Patentierbarkeit von Softwareinnovationen, 2001; PbT Consultants: The Results of the European Commission Cunsultaion Exercise on the Patentability of Computer Implemented Inventions, 2001; and Hart, Robert – Holmes, Peter – Reid, John: The Economic Impact of Patentability of Computer Programs, 2000.

¹⁹ See http://swpat.ffii.org/papers/ewhc-mh020315/index.en.html

Let's discuss each of these proposals in turn. First, solving the problem of trivial patents may not help at all. While much of open source code may be classified generic there are major projects, which produce new state of the art (operating systems, databases). Also proponents must quite optimistically assume that patent system works efficiently in software development and, if it does not, it can be easily fixed. However, numerous studies have shown that the function of the patent system in software industry has little or nothing to do with innovative activity. Rather the patent system may benefit the marketing, financing, litigation and global business strategy of large companies.²⁰

Any suggestions that open source developers should use the patent system have more fundamental problems. First, if GNU licenses are used then patents may have no value: these licenses require all patents relating to GNU licensed software to be licensed to anyone free of charge. As noted above, also other open source licenses have similar terms. It may prove practically impossible to change such a well-laid principle in all licenses whose rationale is strongly ethical.²¹ Second, for a patent pool in open source world to be functional, it should be comparably as strong as the patent portfolios of big IT companies. It is difficult to think hobbyists and small companies to build a strong patent portfolio consisting of thousands of patents anytime soon.²²

Next, while adjusting software development to the model assumed by patent law may function in large companies it can hardly fit the informal and distributed nature of open source development model. It should be also noted that even large companies depend on external help with patents.²³

http://features.linuxtoday.com/news_story.php3?ltsn=1999-05-16-003-05-NW-LF ²² For example IBM and Microsoft apply currently for over 3000 patents annually. See "Gates wants patent power", News.com, July 29, 2004.

²⁰ See e.g. Bessen, James – Hunt, Robert M.: An Empirical Look at Software Patents, National Bureau of Economic Research Working Paper #2003-17 and David, Paul A. – Foray, Dominique -- Hall, Bronwyn H. -- Kahin, Brian -- Steinmueller, W. Edward: "Is there really a good economic rationale for an EU Directive on Software Patents?", working paper 14 July 2003.

²¹ See also Stallman, Richard: "Saving Europe from Software Patents"

²³ Blind – Edler – Nack – Strauss (2001).

Finally, Blind et al. propose that in the future it should be considered whether non-commercial use of open source software should be exempted from patent claims – even if it takes place within commercial environment. This approach has its obvious advantages. First of all, it would be a commonly accepted social policy. Second, and perhaps more importantly, open source development as a methodology would not need to adapt to the abstract rules of law, which may be far from the software development reality. Instead, laws would be adapted according to the winning development method.

6. Liability Exceptions for Open Source?

Let's think in practical terms for a moment. Individual developers and small companies have good arguments that the open source development model would suffer from software patents. Their argument has its roots much deeper than the current public debate over the proposed directive. Their argument does not stop with the directive.

Since open source forms an essential and growing part of the software industry today, it is in the interest of all participants, big global companies alike, to find a solution to the worries of patent opponents. Arguing that the patent system will work in the future and increase innovation in the software industry is not an answer to their problem. The question is about open source development and licensing model, which is in conflict with the software development and licensing model assumed by patents.

In this article, we have formed a practical approach to the problem of open source and software patents. What realistic proposals do we have for those affected by patents? From individual developer or small company perspective, the insurance option is worth considering. The annual price of the insurance reflects the actual price of software patents. From social policy perspective, writing new exceptions in a new law seems like the best possible answer. Unfortunately the political debate in the EU has been going forward and backwards without clear answer to those worried about open source. Currently, the exception approach is at risk.²⁴

²⁴ Initially, the commission proposal didn't include too many exceptions for open source. EU parliament changed the tone by voting for numerous exceptions to the directive

Finally, a quick speculation is in place. What would happen if the exception approach would win in the end? Quite interestingly, Blind et al go on to note that if a liability exception to open source is sought then also the TRIPS agreement would need similar revisions. Here, one may compare the position of open source advocates to those of developing countries. Both have general intellectual property (software code and indigenous cultural knowledge), which multinational corporations try to claim as their private property. The difference is that open source advocates work mainly in the developed world and their relative economic and political power may be far greater than those of the third world developing countries. Open source advocates, if united to oppose TRIPS, may actually become a serious player in the fight of the ownership of the knowledge economy.²⁵ Now, wouldn't that be interesting?

proposal in the fall 2003. In spring 2004, Ireland's new "compromise proposal" has once again revised most of these changes out. The debate goes on.

²⁵ For a recent overview of the issues, see Drahos, Peter – Braithwaite, John: Information Feudalism: Who Owns the Knowledge Economy?, Earthscan Publications, 2002