SCRIPT-ed

Legal Challenges to Open Source Licences

Andrés Guadamuz González[*]

Abstract

This paper will concentrate on presenting a legal analysis of two of the main challenges to open source software: SCO's litigation and software patents. The paper discusses the validity of such challenges, their possible impact to the future of open source software, and the possible legal defences used against them.

---

[*] Co-director, AHRC Centre for Studies in IP and Technology Law, University of Edinburgh.

I. Introduction

It is practically unnecessary to point out that non-proprietary software licences have been generating a tremendous amount of interest in recent years. Both the Free Software and Open Source Software (henceforth FOSS) definitions provide a general framework for what non-proprietary licences should cover and the minimal rights that should be granted to users of products licensed through with these methods.[1]

At the same time, a growing number of software projects have been shown to use FOSS licences.[2] The vitality of the licensing model can also be seen from the fact that Sourceforge – the largest open source software repository – lists almost 60,000 software projects that use a FOSS licence approved by the Open Source Initiative (OSI). The popularity of the novel licensing model has prompted a number of legal studies that analyse the validity and/or enforceability of existing licences,[3] as an eventual problem with the legal principles underlying the documents could lead to a disaster for the nascent open source software industry. The preliminary legal analysis of FOSS licences conducted in the literature seems to suggest that the model is legally valid, a fact that lends substantial credit to the movement. Other indications as to the soundness of the licences are also encouraging; a recent ruling in Germany that has recognised the validity of the General Public License (GPL),[4] which further serves to stress that FOSS is a global phenomenon that is revolutionising the entire field of software development.

However, complacency at this stage would be seriously misplaced. Although there are many positive signs that point towards the health of the licensing system, there are also a number of developments that could spell trouble in the future, and have received less coverage in the literature. These new challenges are often prompted by more aggressive tactics used by some proprietary software developers and other adversaries of the FOSS development mindshare. The tactics largely include litigation and the use of so-called Fear-Uncertainty-Doubt (FUD) tactics designed to undermine the popular perception of the open source philosophy. There are also challenges that are prompted by potential weaknesses in the licences that have not yet been explored in the limited existing cases. Another relatively underreported problem for the future of FOSS is the growing threat coming from software patents.[5]

---

[1] LE Rosen, *Open source licensing: software freedom and intellectual property law*, (2004).

[2] M O'Sullivan, "Making Copyright Ambidextrous: An Expose of Copyleft" (2002) 2002 Journal of Information, Law and Technology 3 <http://elj.warwick.ac.uk/jilt/02-3/osullivan.html>.

[3] See: R Gomulkiewicz, "De-Bugging Open Source Software Licensing" (2002) 64 University of Pittsburgh Law Review 75; A Guadamuz, "Viral contracts or unenforceable documents? Contractual validity of copyleft licenses" (2004) 26 European Intellectual Property Review 8 331; C Nadan, "Open Source Licensing: Virus or Virtue" (2002) 10 Texas Intellectual Property Law Journal 349; and D Ravicher, "Facilitating Collaborative Software Development: The Enforceability of Mass Market Public Software Licences" (2000) 5 Virginia Journal of Law & Technology 11.

[4] J Höppner, "The GPL prevails: An analysis of the first-ever Court decision on the validity and effectivity of the GPL" (2004) 1 SCRIPT-ed 4 662 <http://www.law.ed.ac.uk/ahrb/script-ed/issue4/GPL-case.asp>.

[5] D Evans and A Lynne-Farrar, "Software Patents and Open Source: The Battle Over Intellectual Property Rights" (2004) 9 Virginia Journal of Law & Technology 10.

This paper will briefly describe the two main threats to the FOSS development, while the last part will look at the possible strategies that can be taken by the community in order to counteract this perceived threat.

II. FUD as Law

The success of open source software both commercially and as a development model cannot be denied.[6] Even those who doubt and criticise open source openly tend to recognise the value of the system to generate results and as a collaborative tool the likes of which have seldom been seen in recent times. This is exemplified by the infamous Halloween Documents.[7] Of particular interest to this article is the fact that FOSS has been particularly successful in the corporate world, where open source systems and applications are viewed as cost-effective, secure and reliable.[8]

The success of FOSS has prompted some opponents of open source to find other ways of undermining the system. One of the most effective ways in which this has been done is by attempting to dissuade potential corporate users of open source software from choosing this solution. Traditionally, FUD tactics have been achieved by issuing dire warnings about security concerns, interoperability, or simple ideological attacks.[9] However, the scare tactics are becoming more sophisticated by making certain legal claims about the validity of FOSS licences. The objective of this tactic is to generate enough FUD that corporate users will simply choose safer options legally-speaking, namely proprietary software. There are many different ways of generating legal FUD, but the most effective way to achieve legal uncertainty is by conducting litigation.

Until recently, there had been no court cases against non-compliance of a copyleft licence, and the few incidents that had arisen had been dealt swiftly with cease-and-desist letters from the Free Software Foundation where those parties suspected of producing proprietary software were warned that they were in breach of their contractual obligations.[10] This all changed when a developer of non-proprietary database software named MySQL sued NuSphere – a software company that it believed was using its source code to produce proprietary software – something that contravened the terms of the GPL.[11] This file was issued in response to a suit filed by NuSphere claiming "*breach of contract, tortious interference with third party contracts and relationships and unfair competition*".[12] This case was settled out of court; hence the GPL did not receive a judicial review in this occasion. However, this

---

[6] S Weber, *The success of open source*, (2004).

[7] E Raymond, *The Halloween Documents*, Open Source Initiative (1998) @: <http://www.opensource.org/halloween/>.

[8] D Wheeler, "Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)?" (2005) <http://www.dwheeler.com/oss_fs_why.html>.

[9] For example, Bill Gates calling open source programmers communists. M Kanellos, *Gates taking a seat in your den*, CNET News (2005) @: <http://tinyurl.com/3kl9s>.

[10] G Moody, *Rebel code: Linux and the open source revolution*, (2002).

[11] K Nikulainen, "Open Source Software: Why is it here and will it stick around?" (2004) 1 SCRIPT-ed 1 149 <http://www.law.ed.ac.uk/ahrb/script-ed/docs/opensource.asp>.

[12] A FAQ about the case can be found here: <http://www.mysql.com/news/article-75.html>.

was only the opening shot in what is set to become one of the largest and most complex legal battles that the software industry has ever seen.

The legal question about the validity of copyleft licensing models broke spectacularly in legal circles in March 2003 when the SCO Group – a well known software developer of UNIX related products – filed a lawsuit against IBM alleging that the company was infringing its intellectual property over the UNIX kernel.[13] The full details of the suit are still sketchy because SCO is keeping some of the most detailed information of the code they allege to have been protecting as a close secret, not letting it be known which part of the code it claims ownership of.[14] However, it is known that SCO claims that back in 1985 AT&T and IBM signed a contract to produce a version of UNIX called AIX. In 1995, SCO purchased all of the intellectual property related to UNIX from AT&T, hence the claim they have filed against IBM. It would seem that SCO is somehow making claims that they own part of the code for AIX, or that they own some other part of the UNIX kernel code that is used in most machines running Linux distributions. Furthermore, SCO threatened to sue every corporate Linux user for copyright infringement,[15] claiming that any Linux user must purchase a licence from them. This threat finally came to fruition in March 2004 when they sued DaimlerChrysler and auto parts retailer AutoZone, two corporate Linux users.[16] As a result of this action, IBM countersued SCO claiming that the company has been infringing its own copyrights and patents, and also alleging that SCO is in violation of the GPL because they are users and modifiers of the Linux kernel, which is licensed with the GPL.[17]

It is too early to ascertain the strength of SCO's arguments, but it has become clear that this case has increased the stakes in the financial importance of copyleft licences, and hence the importance in making sure that the licence terms are valid.[18] Nevertheless, SCO's arguments should be met with considerable scepticism given the nature of the development of UNIX and Linux. It will be very difficult for SCO to prove ownership of some code that was developed under an atmosphere of collaboration, and that may date as far back as 1969. In fact, some of the code that they have finally displayed has been shown to be of dubious origin.[19] Another aspect is that one should assume that the timing in this case counts. Why did SCO wait until now to exercise their intellectual property rights? Could this have to do with the fact that SCO's share price has quadrupled since this case made the headlines?[20] One also

---

[13] Caldera Sys., Inc. v. Int'l Bus. Machs. Corp. (D. Utah 2003) (No. 03-CV-0294).

[14] J Harvey and T McClelland, "SCO v. IBM: The Open Source benefits and Risks are Real" (2003) 20 Computer & Internet Lawyer 9 1.

[15] P Galli, "SCO Warns Linux Users of Legal Liability", E-Week, (May 14, 2003), @: <http://www.eweek.com/article2/0,3959,1149623,00.asp>.

[16] T Weiss, "SCO Sues Two Linux Users, Warns About Further Action," Computerworld (March 8, 2004), @: <http://www.computerworld.com/softwaretopics/os/story/0,10801,90868,00.html?f=x72>.

[17] KD Goettsch, "SCO Group v. IBM: The Future of Open-Source Software" (2003) University of Illinois Journal of Law, Technology & Policy 581.

[18] The most recent developments in this case can be followed @: <http://www.groklaw.net>.

[19] J Lyman, "SCO's Evidence Raises Questions About Case" E-Commerce Times (20 August 2003), @: <http://www.ecommercetimes.com/story/31386.html>.

[20] For a chart of SCO's stock prize, see: <http://stocks.tradingcharts.com/>.

must be suspicious of the fact that Microsoft has obtained a Unix licence from SCO, which has prompted repeated accusations that SCO may be a pawn in Microsoft's attacks against FOSS.[21]

Although it would be foolish to predict what will happen in this case, it must be said that SCO's claims look flimsy. If SCO's intention is to win the case, it seems likely that they will fail. If their intention is to generate doubt and uncertainty in potential FOSS users, then the answer could be that they may be succeeding. In an already risky and difficult corporate environment, companies are attempting to minimise their potential liabilities. One way of doing it is by making sure that the company will not be sued, or having sufficient insurance in case that this takes place. The problem with litigation such as the SCO case is that it creates the false impression in the corporate world that FOSS stands on shaky legal ground, and that the potential liabilities are just too great. This may already be taking place, as lawyers, consultants and insurance companies, take a "wait and see" approach towards FOSS and warn against the use of applications and operating systems based on some sort of open source or free software licence, which will only serve to steal momentum from the movement.[22]

III. Software Patents

The problem of the patentability of software has become one of the most debated issues with regards to open source software. This is because FOSS licences rely heavily on copyright protection. This has been turned on its head in the United States after the United States Patent Office has been increasingly awarding patents for software, something that was not possible at earlier stages.[23] The change has taken place for many reasons, but one often cited justification is that copyright law in the United States has had problems with software protection and the applicability of the idea-expression dichotomy found in American copyright doctrine, problems expressed in the clunky doctrine of so-called Abstraction-Filtration- Comparison.[24]

Europe has usually been considered as a different environment altogether for software patentability because the European Patent Convention specifically lists software as non-patentable matter.[25] However, practice and case law have allowed a limited patentability of "computer implemented inventions" that involve a technical effect (or contribution, or process).[26] These cases have allowed for a limited patentability threshold to exist as long as the invention that is going to be implemented through a computer fulfils this requirement of technicality.[27] The problem since the first rulings of the Technical Board of Appeal of the European Patent Office has been precisely to

---

[21] T Olavsrud, "Microsoft Buys Into SCO Group's Unix," Internet News (May 19, 2003), @: <http://www.internetnews.com/dev-news/article.php/2208691>.

[22] For example, see this short piece by corporate lawyers: R.L. Meyer and H. Stewart, "The Risks of Open Source Software", FindLaw, 2003, @: <http://tinyurl.com/4r76b>.

[23] K Nichols, *Inventing software: the rise of "computer-related" patents*, (1998).

[24] Found in Computer Associates International, Inc. v. Altai, Inc., (2nd Cir. 1992) 61 USLW 2434.

[25] Specifically, Art. 52(2) says that software is not an invention, and therefore it is not patentable.

[26] Cases such as VICOM [1987] 2 EPOR 74; Merrill Lynch [1989] RPC 561; Gale [1991] RPC 305; Fujitsu [1997] RPC 608.

[27] R Widdison, "Software Patents Pending?" (2000) The Journal of Information, Law and Technology 3 <http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2000_3/widdison/>.

define exactly what is meant by technical effect. It is well understood that source code, or the literary and textual element of software, cannot be patented, but that if the software produces some sort of effect in the same way that an invention does, then it will be awarded protection.

Even though the application of the technical effect principle has been rather muddled, there appears to be a common element that the contribution must make a considerable contribution to the prior art.[28] However, even with this element in common in most of the existing rulings, the application of these principles in real life has been uneven, as is often the case with vague and ill-defined legal concepts. This has prompted the European Union to propose a directive to harmonise different aspects about the patentability of software related inventions.[29] This proposal was set to be a simple overhaul of European patent practices to make the wording of technical effect more precise.[30] The actual definition of what constitutes a technical contribution is similar to the requirements of prior art encountered in the case law. The novel approach of the proposal is that it offers a definition of patentable inventions that provide this technical effect. The text defines a computer implemented invention as that in which the invention "*involves the use of a computer, computer network or other programmable apparatus and having one or more prima facie novel features which are realised wholly or partly by means of a computer program or computer programs …*"

This would seem straightforward, but the proposal has met with fierce opposition from software developers, particularly FOSS developers. This is mostly because the American system is often seen as too broad, allowing patentability of all sort of software that have been around for a long time.[31] FOSS developers see this as a threat to the fact that FOSS licences rely heavily on copyright. There is also a sense that this is a slippery slope, and that American-style broad patents will be awarded in Europe even with the new wording, something that is already taking place, even with the existing restrictions.

This can be illustrated in many ways, but one example will help to make the point clearer. U.S. Patent 6,330,551 has been awarded to protect automated online dispute settlement systems. The central part of the patent is that it protects a system akin to blind-bidding, where opposing sides in a dispute place their monetary claims to a computer system (preferably an online secure system) where the computer calculates the claims and eventually reaches an automated settlement that fulfils the stated expectations of each of the participants. The problem with this patent is that it seems to be a very obvious application of blind-bidding technology, for which there is significant prior art.[32] As such, it should not be subject to the alleged high-standard of

---

[28] For example, the Merril Lynch case says that "There must be some technical advance on the prior art in the form of a new result."

[29] Proposal for a Directive on the Patentability of Computer-implemented Inventions COM(2002) 92.

[30] A Duffus, "The Proposal for a Directive on the Patentability of Computer-implemented Inventions" (2002) 16 International Review of Law, Computers & Technology 3 331.

[31] See for example, US Patent No. 6865546, which protects a system that determines a buyer's age according to previous buying records.

[32] See: RM Issaac, "Theories and Tests of "Blind Bidding" in Sealed-Bid Auctions" (1989) 20 RAND Journal of Economics 2 214; and GR Shell, "Computer-Assisted Negotiation and Mediation: Where We Are and Where We Are Going" (1995) 11 Negotiation journal 2 6.

patentability of software inventions in Europe, but the fact is that this "invention" has also been given a patent in the UK (GB2345997).

Cases like this one abound, and it is why the proposed text of the directive is not effective. It must be mentioned that this draft directive has had a tortuous road towards approval and implementation,[33] with the European Parliament and the European Commission clashing about the exact wording of the definition of technical contribution. At the moment of writing this paper the fate of the directive is still not clear, but it is to be hoped that if the directive is finally approved, that the text will be as precise and detailed as possible to diminish the possibility of problems with the application of the concepts.

What is clear is that if there is an increase in the patenting of software ideas, this would have a detrimental effect on FOSS development. As the licences rely heavily on copyright, it would be possible that developers will find themselves with unenforceable licences because they only cover the copyright aspects of the software. It is also problematic because FOSS programmers could be subject to patent infringement cease-and-desist letters, or even lawsuits, resulting from an overly-brad patent that should not have been granted in the first place. Although developers could try to get these patents struck down in court, patent litigation is very expensive,[34] and small and medium FOSS projects will not have the resources to fight such battles. It would be possible that patents could be used by proprietary software companies as another armament in their FUD arsenal.

IV. Defending the FOSS Model

The implications of the legal challenges described in the two previous sections for the field of FOSS development cannot be understated. An adverse ruling in the SCO case would be catastrophic to the entire movement, as would be a widespread legal infringement suit brought against open source firms or small programmers. Can the open source community do something to counter these threats? There may be some ways to do it.

Perhaps the best way to make sure that FUD is not used is already being done effectively by maintaining a strong community that is prepared to use the extensive powers of the internet as an information dissemination tool to effectively counter the arguments made against the community. Issue-oriented blogs and websites such as Groklaw and the Foundation for a Free Information Infrastructure (FFII) provide excellent examples of what organised (and even disorganised) web activism can achieve. Another strategy against FUD is for academics to become more involved and write more scholarly works to counter the many inaccuracies that are often displayed in a lot of the attacks.[35] Still, the best defence is to make sure that the licences are strong and valid, and that they are geared towards solving these problems.

---

[33] AWS Williams, "European Commission: proposed Directive for patents for software related inventions" (2004) 26 European Intellectual Property Review 8 368.

[34] A Jaffe and J Lerner, *Innovation and Its Discontents*, (2004).

[35] JB Wacha, "Taking the Case: Is the GPL Enforceable" (2005) 21 Santa Clara Computer & High Technology Law Journal 451.

In the area of patents, there are serious problems with the existing reliance of the licensing model on copyright, so there is the need to redraft FOSS licences to accommodate patents. . Steps are being taken in some instances to minimise the possible threat of software patents to the entire model, particularly through the inclusion of so-called "patent clauses" to some licences as is the case with the GPL.[36] The GPL clause states that "*we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.*" Other recent licences, such as the Apache Licence (version 2.0), contain a patent assignment clause that not only allows users to use the copyright part of the software, but that it also assigns patent claims that arise from the protected software.

Recent developments have suggested that there may be other ways of protecting FOSS from software, and that strict licences are not needed to provide a common pool of patents that can be used for open source purposes. IBM has made the headlines of every major technology-related publication by stating that it will not enforce 500 software patents that it owns if they are used by open source software projects.[37] This unprecedented move has been achieved through a clever use of contract law. IBM has published a legally-binding promise not to enforce a number of their patents to those software projects that are released to the public through a licence approved by the Open Source Institute. This element of IBM's pledge is very important, as it gives a tight definition of what will be an open source project. The definition reads:

> "*Open Source Software is any computer software program whose source code is published and available for inspection and use by anyone, and is made available under a license agreement that permits recipients to copy, modify and distribute the program's source code without payment of fees or royalties. All licenses certified by opensource.org and listed on their website as of 01/11/2005 are Open Source Software licenses for the purpose of this pledge.*"[38]

The document goes on to promise that IBM will not assert any of the listed patents in the united States, or its counterparts worldwide, against open source projects, defined as above. The document ends with a list of the 500 patents subject of the promise. This announcement should be met with some scepticism, as IBM has a considerable software patent portfolio, and was awarded more than 3,000 patents in 2004 alone.[39] One should also be sceptical about the possible legal validity of such promise, as the issue of unilateral promises varies from one jurisdiction to another.[40]

---

[36] M Valimaki, "A Practical Approach to the Problem of Open Source and Software Patents" (2004) 26 European Intellectual Property Review 12 523.

[37] "IBM frees 500 software patents", BBC News (11 January 2005), @: <http://news.bbc.co.uk/1/hi/technology/4163975.stm>.

[38] IBM. IBM Statement of Non-Assertion of Named Patents Against OSS, (1 January 2005), @: <http://www.ibm.com/ibm/licensing/patents/pledgedpatents.pdf>.

[39] "IBM, Matsushita, Canon and HP received the most US patents in 2004", IT Facts (2005), @: <http://www.itfacts.biz/index.php?id=P2370>.

[40] J Gordley (ed) *The Enforceability of Promises in European Contract Law*, (2001).

V. Conclusion

These two challenges to open source software that have been described are well known in open source circles, with many different experts in the field providing some possible solutions to what could be a problematic situation in the near future. This paper has attempted to provide some room for the debate and to suggest some few ways in which the threats can be avoided, or at least minimised. However, there may be little that FOSS developers and academics can do to decrease the dangers in the future until there has been some legislative action in order to make the area of software patents a more coherent field. Perhaps the best solution at the moment is for academics to continue to warn about the hazards of the status quo and educate FOSS developers of the challenges ahead.