SCRIPT-ed

Volume 1, Issue 1, March 2004

OPEN SOURCE SOFTWARE: WHY IS IT HERE AND WILL IT STICK AROUND?

Kimmo Nikulainen[*]

Abstract

This paper discusses the details behind the Open Source Software development scheme by looking at some of the most prevalent licences, and then by asking the question of whether Open Source will survive the legal battles that are starting to affect it, in particular the SCO v IBM case.

DOI: 10.2966/scrip.010104.136

[*] LL.M; University of Edinburgh

0. Introduction

This paper is divided in three sections. In the first, the history and ideology behind the Open Source movement, and in a larger sense, the Free Software movement is discussed. A fundamental question that must be asked is the one of motivation behind the creation of a product that apparently offers no financial incentive. Thus, the traditional incentive theory that is seen to justify intellectual property rights is questioned in the second part. The commons model on the one hand and the proprietary model on the other, are compared and it will be suggested that even though the most vocal proponents of open source software (hereafter OSS) and free software (hereafter FS) might suggest otherwise, the terms "open" and "free" are misnomers in a legal sense in this context. The latter half of discussion is devoted to the topical and important question: will open source based software be a viable business strategy in the future? This question will be examined in the light of recent lawsuits initiated by Unix licensor SCO in the United States. It is suggested that despite some healthy scepticism and due care on behalf of certain legal consultants[1], open source will continue to flourish and is likely to grow in importance in a wide sphere of industries.

I. OSS/FS in general

a. Ideology behind the movement

The history of OSS is strongly connected to that of hackers and the Internet. Massachusetts Institute of Technology was one of the first hacker communities where modern English terms like "hacker" and "hacker ethics" were coined in the 1960s. In the early 1970s several universities in the United States had joined in the United States government's project that connected computer networks together, called ARPANET that was already, due to its inter-network quality, called the Internet at that time. There were discussions about selling the Internet to a commercial enterprise but the universities were strongly opposed to such plans and it remained independent. The Internet has been built from its early days by a voluntary work force including university staff and students who shared and developed utilities that were used in the network.[2]

One of the cornerstones of hacker ethics is that sharing information is a virtue and there exists a moral duty to disseminate one's knowledge in form of computer programs and further develop the infrastructure where the programs are used. The early hackers in MIT remained true to this moral rule and allowed their software to be freely copied and distributed, and always made the source code available for their fellow members of the computing community.[3]

---

[1] See for example US law firm Bingham McCutchen's Commercial Technology Alert, 3 July 2003, straightforwardly suggesting that "notwithstanding the perceived initial cost-savings in using open source software, the potential risks of infringement claims and loss of intellectual property rights in proprietary software lead us to recommend against such use…". @: <http://www.mccutchen.com/Bingham/updates.asp>

[2] J Sandred, *Managing open source projects* (2001), 9

[3] Id

Of all the departments at MIT, the Artificial Intelligence Lab was especially known as the cradle of hacker culture. That is where Richard Stallman – who would soon play a major part in the growth of OSS/FS movement – also worked in the 1970s and 1980s. Stallman opposed the commercialisation of the universities' computer software; he felt that even though there were strong standards in the UNIX markets, they would be used to the benefit of the proprietors as opposed to the community at large.[4]

b. First there was Free Software

In 1984 Stallman left the MIT and started a project aiming to build a freely available operating system that he called GNU (GNU's Not Unix). The following year he founded a non-profit organisation called Free Software Foundation (FSF), which promotes free dissemination of software and develops the GNU.[5]

Stallman calls his software "free" which is understandably ambiguous and easy to misinterpret. Stallman emphasises that free in his context refers to freedom, not price.[6] Stallman includes four freedoms in his definition:[7]

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbour (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

Stallman noticed that he needed something to protect the freedom of his software, since making it available in the public domain would have allowed anyone to include it in their derivative, proprietary software, very much the opposite from the goal he wished to achieve. The answer to his problem was "copyleft". Copyleft will be further discussed below[8] but this is an appropriate place to lay out Stallman's idea.

> *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.*[9]

---

[4] Id, 14

[5] D Bretthauer, "Open source software: a history". @: <http://www.lita.org/Content/NavigationMenu/LITA/LITA_Publications4/ITAL__Information_Technology_and_Libraries/2101_Bretthauer.htm>

[6] GNU Project, "Why free software is better than 'open source'". @: <http://www.gnu.org/philosophy/free-software-for-freedom.html>

[7] GNU Project, "The free software definition". @: <http://www.gnu.org/philosophy/free-sw.html>

[8] See note 62 below

[9] GNU Project, "What is copyleft?" @: <http://www.gnu.org/copyleft/copyleft.html>

Stallman's tool of choice to realize his copyleft ideal was a new kind of software licence called GNU General Public License (GPL), which was first published in 1989 and the second, currently used version in 1991.[10] The various licences pertaining to OSS/FS are discussed below.[11]

c. Enter Open Source

In 1997 a group of FS developers were discussing about less radical ways of furthering their cause – something that would catch the eye of the corporate world. They wanted to find a new term for their method, and another MIT graduate Christine Peterson coined the term "Open Source Software". The group founded a non-profit organisation called Open Source Initiative (OSI), aiming to find more commercial justifications to free software.[12]

Open Source Definition (OSD) is OSI's definition of open source software and more liberal than the one by FSF. The most notable difference is the fact that OSD allows software to be included more freely in proprietary programs. OSI and FSF make an interesting couple: they disagree on the basic principles, but agree on the practical recommendations. They work together and do not consider each other an adversary.[13]

In order for a software licence to qualify as open source, the conditions of The Open Source Definition must be met. OSD contains the following provisions:[14]

- Free Redistribution: The first of the three key elements of OSD.[15] An OSD license may not restrict any third party from including the software as a part of their product.
- Source Code: The second key element. The program must include source code, and must allow distribution in source code as well as compiled form.
- Derived Works: The third key element. The license must allow modifications and derived works.
- Integrity of The Author's Source Code: The author of a particular piece of code must be recognized. This could be seen as the fourth key element of OSD as credit-giving is imperative for the Open Source movement to succeed. One has to keep in mind that hackers spend countless hours programming modified versions of software because of the prestige they will enjoy in their community.[16]

---

[10] GNU Project, "The GNU General Public License". @: <http://www.gnu.org/licenses/gpl.html>

[11] See note 18 below

[12] Open Source Initiative, "The open source case for hackers". @: <http://www.opensource.org/advocacy/case_for_hackers.php>

[13] Id

[14] Open Source Initiative, "The open source definition". @: <http://www.opensource.org/docs/definition.php>

[15] D Kennedy, "A primer on open source licensing legal issues: copyright, copyleft and copyfuture". @: <http://www.denniskennedy.com/opensourcedmk.pdf>

[16] R W Gomulkiewicz, "How copyleft uses license rights to succeed in the open source software revolution and the implications for Article 2B" (1999) *36 Houston L.R.* 187

- No Discrimination Against Persons or Groups. The rationale of including this provision is to guarantee the widest possible pool of contributors for the open source pool.
- No Discrimination Against Fields of Endeavour. The reason behind this provision was to prohibit license traps that could exclude commercial users from joining the community.
- Distribution of License. The rights attached to the software must apply to everyone to whom the software redistributed. This clause is purported to prohibit an indirect closing-up of software by means of a non-disclosure contract.
- License Must Not Be Specific to a Product. If the software is first distributed as a part of larger package and then extracted from that package, it will still enjoy the original privileges under the original license. The commercial importance of this clause is that it prohibits the exclusion of a program from one of the several distributions of large OSS bundles, such as Linux operating system that is being distributed by several companies.
- The License Must Not Restrict Other Software. If an open source program is distributed as a part of a large bundle, there may be no restrictions on other parts of the bundle, they may well be traditional, closed programs.
- The License Must Be Technology Neutral. This clause was added to tackle click-wrap licenses that could conflict with certain distribution methods. "Conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-graphical environment that cannot support popup dialogues."[17]

d. Various licence types

There is a vast number of different open source licences to choose from. The licences can be categorized in three classes: unrestrictive, restrictive, and highly restrictive.[18] The unrestrictive licences such as the BSD family allow the creation of proprietary works from the original code. In this way an open source originated product can end up becoming closed. The restrictive licences, on the other hand, purport to keep the derivative code tightly in control – and ensure that the future works also remain open. The most popular highly restrictive licence is Stallman's original GPL, or General Public License that effectively puts his copyleft ideology in operation.[19] A popular example of mid-ground, restrictive licence is the Mozilla Public License (MPL) that was created by Netscape.

---

[17] See Section 10 of the open source definition @: <http://www.opensource.org/docs/definition.php>

[18] J Lerner and J Tirole, "The Scope of Open Source Licensing, National Bureau of Economic Research Working Paper No. w9363, December 2002". @: <http://papers.nber.org/papers/W9363>

[19] See note 8 above. A useful list of various open source licenses is available on GNU Project's home page @: <http://www.gnu.org/philosophy/license-list.html>

1. The General Public License (GPL)

The main features of the GPL are persistence and viral effect. It is said to be persistent because all the derivative works[20] must be licensed with the same GPL. In other words, the creator of a derivative work cannot gain an advantage over the creator of the original, GPL licensed work in this respect.[21] The viral effect means that if a piece of GPL code is used in a work that includes proprietary or "closed" code, then this resulting program must become open under the GPL in its entirety. The question of viral effect has caused a healthy amount of discussion and Microsoft has expressed its concern regarding the potential demise of its business model. In the worst possible case scenario a tiny piece of GPL'd code contaminates an entire software bundle, such as a future version of the highly successful Windows operating system, rendering it legally impossible to hide its source code.[22]

Some notable shortcomings can be found in the GPL compared to other popular open source licenses. There are no provisions concerning the choice of law or choice of forum. The general terminology is ambiguous and whole concept rather propagandist. This is undoubtedly due to its creator's idealism and one of the reasons why the corporate world has been relatively slow to jump on the open source bandwagon. The lack of any indemnity clause in case of possible breach of a legally enforceable interest also adds to the uncertainty.

2. Mozilla Public License

The Mozilla Public License (MPL) was used to open source Netscape's browser software in 1998. Netscape could not release the source code of their popular web browser under the GPL because it incorporated software licensed from other developers.[23] Furthermore, Netscape being a commercial organisation, it wanted to make sure that in the future it would still be able to earn revenue from the browser and future developments of the browser. As a result, Netscape decided it was best to produce its own licence, The Mozilla Public License.

The MPL is a more legalistic license than the GPL, which is more of an ideological declaration than a commercial software licence that the MPL clearly resembles. Under the MPL it is legal for a business to charge a fee to take on liability for the software, and to support it. Such provisions in the license make it possible for a business to create value added services such as consultancy, installation, debugging and so on. The MPL allows additions to the software to be licensed on different terms from the MPL, meaning that the MPL does not have the viral effect of the GPL.[24]

The core idea of the MPL is that the original creator of code has more rights toward the work than the developer society does. The licence is persistent as far as the source

---

[20] It should be noted that the English term "derivative work" pertains to US law, whereas the UK counterpart is "adaptation"

[21] M Välimäki, "Avoimen lähdekoodin ohjelmistolisensseistä" 2002 *Defensor Legis* 851, 854

[22] Microsoft's concern seems exaggerated since there are several commercial open source licenses that allow mixing with proprietary software, and in any case they could always write a piece of software themselves if they were suspecting it being GPL'd

[23] R Chapman and A Rose, "Open source software licensing parts I & II" Feb 2003

IT Law Today 19

[24] Id

code is concerned but the original creator reserves the right to use a different licence for the object code on a later date. The described technique is based on the fact that source code and object code of a computer program exist for two separate purposes: source for programmers and object for end users. Therefore the licensing terms can vary between the two.[25]

3. BSD licenses

Used originally in the BSD Unix (Berkeley Software Distribution) bundle, this unrestrictive licence has been widely adopted by the open source community. The most popular variants are MIT, X and Apache.[26] They not only allow usage, modification, copying and distribution of software but also make it possible for the coder of a derivative work to alter the licence provisions of a derivative work. Thus BSD licenses are neither persistent nor viral in effect.

A BSD license can be a commercially viable option if it is desired to achieve a large pool of third party developers to create commercial products that support the original product, or are based on the original software. The danger obviously lurks in the fact that a third party may "hijack" the source code and develop its own closed package. Therefore a BSD license is normally recommended to be used only in such components of a software bundle that support the main program and to which the developer does not want to allocate future programming resources.[27]

II Why do Open Source Projects Exist?

a. Possible economic and technical advantages

The use of open source is argued to bring several benefits in comparison to proprietary developing models.[28] These advantages can be economic, e.g. the reduction in research and development costs on the one hand, and technical i.e. resulting in better quality of code, on the other. The traditional argument is that open source offers the software developer increased reliability: it has been said that "given enough eyeballs, every bug [in the code] is shallow".[29] This means that given a large enough base of beta-testers and co-developers, practically every problem in the code can be solved by the time of release. This is not always the case in traditional, in-house programming team constructed software packages.

The OSS community admits that there can be valid cases for keeping the code closed. If considerable research and development effort is required to produce a product based on a new technology, then keeping the code closed would seem to be a plausible business strategy. They are quick to remind, however, that once a competitor comes up with a similar product, one should go open source in order to

---

[25] M Välimäki, "Avoimen lähdekoodin ohjelmistolisensseistä"2002 *Defensor Legis* 856

[26] Id, 855

[27] Id, 856

[28] Unfortunately the spatial limitations of this paper do not allow an elaborated discussion but see note 107 below

[29] This is what E S Raymond calls "Linus's Law" in his publication The cathedral and the bazaar. @:

<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s04.html>

encourage the best pool of co-developers to invest time in it and thus enhance the life cycle of the product.[30]

b. What motivates people to write "free" software

Benkler asks the question of the economic sceptic: why would people continuously work on projects that they cannot claim intellectual property rights over?[31] A well known supporter of open source software, Professor Moglen has a simple answer:

> *So, in the end, my dwarvish friends, it's just a human thing. Rather like why Figaro sings, why Mozart wrote the music for him sing to, and why we all make up new words: Because we can. Homo ludens, meet Homo faber. The social condition of global interconnection that we call the Internet makes it possible for all of us to be creative in new and previously undreamed-of ways.*[32]

Moglen formulated a theory called "Moglen's Metaphorical Corollary to Faraday's Law"[33] that says that if you wrap the Internet around every person on the planet and spin the planet, software flows in the network. Moglen tells a fascinating anecdote to back his view. Microsoft's confidential competitive analysis of Linux that was leaked to Eric Raymond in 1998, and is known as Halloween Memorandum includes a testimony by Microsoft's employee who bought and installed a Linux system on his desktop.[34] After experimenting with Linux for a while he wrote, "The feeling was exhilarating and addictive" about a possibility to see the source code and potentially fix a piece of it to meet his personal requirements.

Lerner and Tirole offer a more detailed analysis on personal incentives to embark on an open source project and start off with an assumption that a programmer only participates in a project, that may be either commercial or open source, if she or he draws a net benefit from the activity.[35] The amount of net benefit is equal to the immediate payoff plus the delayed payoff. The immediate payoffs can be monetary if a programmer works for a commercial firm but more typically in an open source context the immediate payoff will be in the form of personal benefit as the programmer fixes a bug in code or customises a program for his or her needs. The programmer also incurs an immediate cost as his or her time is spent on working on the project, thus rendering the programmer unable to take on an alternative project.

---

[30] H E Pearson, "Open Source – The Death of Proprietary Systems?" (2000) 16 *Computer Law & Security Report* 152

[31] Y Benkler, "Coase's penguin, or, Linux and the nature of the firm" (2002) 112 *Yale Law Journal* 369, 423

[32] E Moglen, "Anarchism triumphant" @: <http://emoglen.law.columbia.edu/publications/anarchism.html>

[33] Michael Faraday was an English physicist who first noticed that electric current flows in a coil of wire that is wrapped around a spinning magnet

[34] The memorandum in question is Halloween memorandum II and can be accessed on Open Source Initiative's web site @: <http://www.opensource.org/halloween/halloween2.php>

[35] J Lerner and J Tirole, "The simple economics of open source" @: <http://papers.nber.org/papers/w7600.pdf> (hereafter referred to as Lerner and Tirole, Simple economics)

The amount of immediate payoff is the current benefit (monetary or personal benefit) minus immediate cost.

According to Lerner and Tirole, delayed payoff would appear more relevant in examining the participant's motivation. They divide delayed payoff into two separate incentives: the career concern incentive and the ego gratification incentive. The former pertains to incentives related to future job offers, shares in commercial open source companies, or access to venture capital markets. The latter is simply related to the value of peer recognition, which Moglen also accepts as a likely incentive for participation in open source projects.[36] These can be grouped together under the heading of signalling incentive. The strength of signalling incentive depends on the visibility of the performance to the relevant audience who can be fellow programmers, potential employers, or venture capitalists.[37]

In comparison with traditional closed source programming work, which offers better immediate payoffs in the shape of salaries and supports the age old economic argument that the prospect of profit encourages innovation, the open source model has some other strengths:[38]

- "Alumni effect": Means that because people who are involved in open source development have already learned relevant programming skills in high school or college, where the principles of open source code are embraced for learning purposes, considerable reductions in development cost can be attained.
- Customisation and bug-fixing: The aforementioned immediate cost in participating in an open source project is lower if it makes the work feel more meaningful by bringing a personal benefit to the programmer in the form of tailoring a piece of code to suit his or her own project.

Lerner and Tirole present three reasons why signalling incentives can divert a programmer from short term benefits and make him or her consider an open source project:[39]

1) Better performance assessment: In an open source world, the developer is exposed to peer review on a level that is not possible to attain in the closed program industry. Because the source code is available for all to see, not only the final product as such can be reviewed, but also the way it was built, i.e. one's work can be scrutinised in respect of the cleverness of one's code and the work can be assessed regarding the actual level of effort needed in reaching the final product.

2) Full initiative: Because in open source the programmer is responsible only to him or herself, as opposed to the boss – on whose interference and guidance the progress

---

[36] In "Anarchism triumphant", Moglen criticises two traditional answers to the question of participant's motivation. He asserts that numerous references to the so called "hacker gift-exchange culture" are wrong because when you live in commons you cannot expect the reciprocity involved in exchange cultures. He concedes that the peer recognition incentive as described by Lerner and Tirole does exist. See @: <http://emoglen.law.columbia.edu/publications/anarchism.html>

[37] This is referred to as "strategic complementarities" by economists, suggesting that programmers will want to work on projects that attract other programmers. See Lerner and Tirole, ch 4.1

[38] Id

[39] Id, ch 4.2

of work would depend - in a traditional firm, the economic theory suggests that the programmer's performance is more precisely measured in the former case.

3) Greater fluidity in labour market: This reason is related to the personal freedom of an individual. It is argued that a programmer has greater freedom of moving on to an alternative project when one does not represent any firm specific values to a company, but is simply one's own master.

c. The friction between IP rights and the digital world

Are IPRs justified in the context of software in the first place? There are influential proponents, such as John Perry Barlow, who have asserted that traditional IPRs are unfit in the information society of the twenty-first century.[40] It is suggested that this is a consequence of an emergence of a true information society which renders digital works unsuitable for the copyright regime that was created for traditional literal works, such as printed books. The famous quote by Stewart Brand neatly sums up this opinion:

> *Information wants to be free because it has become so cheap to distribute, copy, and recombine -- too cheap to meter. It wants to be expensive because it can be immeasurably valuable to the recipient. That tension will not go away. It leads to endless wrenching debate about price, copyright, "intellectual property", the moral rightness of casual distribution, because each round of new (technological) devices makes the tension worse, not better.*[41]

However true the above may be, it has been pointed out that copyright has survived previous technological advancements such as moving pictures and television, and that while it has to be appreciated that use of the Internet does indeed cause problems in respect of speed and scale, these obstacles still fail to make the very core of the copyright system obsolete.[42] A relatively recent phenomenon is the patent protection of computer software; this is more prevalent in the United States than in Europe because the definition of patentable technology is more narrow in Europe.[43] A good reference point can be found in the recitals to the much debated European Copyright Directive as well. The recital recognises the need to adapt to new economic realities but states that no new concepts for the protection of intellectual property are needed.[44]

---

[40] J P Barlow, "Economy of ideas – selling wine without bottles on the global net" @: <http://www.eff.org/~barlow/EconomyOfIdeas.html>

[41] V Crosbie, "Information wants to be free (or does it?)" @: <http://www.clickz.com/design/freefee/article.php/1378891>

[42] H Wiese, "The justification of the copyright system in the digital age" (2002) 12 *E.I.P.R.* 387, 389

[43] P Lambert, "Copyleft, copyright and software IPRs: is contract still king?" (2001) 11 *E.I.P.R.* 165

[44] For a discussion on implementing the EU Copyright Directive in the UK, see Foundation for Information Policy Research @: <http://www.fipr.org/copyright/eucd_intro.html>

d. The commons model & proprietary model

1. Commons model

In the theory of intellectual property rights there is an on-going battle of words between the supporters of the traditional propriety model which, when functioning at its best can discourage inefficient co-operation and result in more innovation and consumer choice, and the commons model whose advocates constantly remind that the immense growth of the Internet is facilitated by open source technologies which are free from the control of any single corporation.[45]

Lawrence Lessig strongly supports the commons approach in his works saying that,

> *We should choose to architect [the Internet] with a commons. Our past had a commons that could not be designed away; that commons gave our culture great value.*[46]

Lessig is suspicious of proprietary standards that have been created by commercial enterprises, such as Microsoft, and believes that the use of open source software in building the cyberspace allows the end user greater control and choice.[47] The most famous example of this school is undoubtedly the GNU-Linux operating system, which embraces the GPL as discussed above. It seems that in the information industries, proprietary control is not necessarily required to encourage innovation[48] and can, indeed, be seen as an adverse force as the existence of a flourishing public domain, where everyone can freely find inspiration for their work is seen as a critical component for technological advancement.[49] The commons model has its strength in the low penetration threshold. Because of the open structure of the Internet, even small firms are able to introduce their innovations to a world wide market.

Weiser has taken a critical look into the failings of the commons model. His initial critique is that the conditions that first facilitated the emergence of the open source based Internet are increasingly not present anymore.[50] In the early days there existed only a small number of stakeholders in the development of the Internet and the software facilitating its basic infrastructure was almost exclusively created by independent, open source communities. Nowadays, however, it seems that an increasing number of proprietary (closed code) developers are prevalent and the creation of common standards will become the exception rather than the norm.[51]

---

[45] R Wagner, "Information wants to be free, intellectual property and the mythologies of control" (2003) 103 Columbia Law Review 995, 995-998

[46] L Lessig, *Code and other laws of cyberspace* (1999), 141 (hereafter referred to as Lessig, Code)

[47] See D G Post, "What Larry doesn't get: code, law and liberty in cyberspace" (2000) 52 *Stanford LR* 1439, 1450. Post takes a critical look into Lessig's influential book. The article is available on-line @: <http://www.temple.edu/lawschool/dpost/Code.pdf>

[48] P J Weiser, "The Internet, innovation and intellectual property policy" (2003) 103 *Columbia Law Review* 535, 570 (hereafter referred to as Weiser, Internet)

[49] Wagner, Information, 998

[50] Weiser, Internet, 573

[51] Id. The World Wide Web Consortium (W3C) that is behind some key standards such as the HTML which is widely used in writing the existing web sites did, however, make a move towards commons strategy, stating that no patented technologies would be allowed in its standards. See W3C's patent policy framework @: <http://www.w3.org/TR/2001/WD-patent-policy-20010816/>

The supporters of commons theory propose that network markets will always choose a single standard and coupled together with this fact, the competition and innovation would be thwarted if intellectual property protection were allowed. Weiser asserts that such assumption of a universal single standard is false because it fails to take into account some important reasons why network competition can occur.[52] These reasons include a situation in a given network where the value of additional customers decline because the network has reached critical mass[53] or alternatively a rival network has managed to catch up the initial advantage of a market leader. It is also pointed out that the supporters of a commons model see the use of open source software as a means of protecting broader public values than just economic ones.[54] The problem with this view is that it calls for a wide scale governmental support which seems unlikely today – a point which has been conceded by the commons camp itself.[55]

2. Proprietary model

Similarly to the criticism that the commons theory camp pour over the traditional closed source software industry, there are some rather vehement commentaries on behalf of the proprietary theory. It does not come as a surprise that Microsoft is among those arguing for the proprietary control model, where intellectual property rights are attached to software.[56] The proprietary control model is based on the theory by economist Joseph Schumpeter that any market power will be temporary because there will always be a new technology that will eventually replace the incumbent.[57] The model is based on the hypothesis that it is in the best interests of the competing firms to invest in research and development in order to gain the market leading position until a rival company takes over with their superior product. Therefore, in this model it appears that monopolies are not only acceptable but necessary for technological advancement.[58] Judge Posner agrees that the Schumpeterian model:

> *We have seen all manner of firms rise and fall in this industry—falling sometimes from what had seemed a secure monopoly position. The gale of creative destruction that Schumpeter described, in which a sequence of temporary monopolies operates to maximize innovation that confers social benefits far in excess of the social costs of the short-lived monopoly prices that the process also gives rise to, may be the reality of the new economy.* [59]

---

[52] Weiser, Internet, 574-575

[53] For a discussion on critical mass, see page 22 of A Bonaccorsi and C Rossi, "Why open source software can succeed" @: <http://opensource.mit.edu/papers/rp-bonaccorsirossi.pdf>

[54] See also note 36 above

[55] Lessig, *Code*, 255

[56] C Mundie discusses the shortcomings of OSS in the data security context @: <http://www.microsoft.com/presspass/exec/craig/06-20softwareecosystem.asp>. In another article he addresses some general criticism on the OSS movement, see @: <http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.asp>

[57] Weiser, Internet, 576

[58] Id, 577

[59] R Posner, "Antitrust in the New Economy" 14 Sep 2000 Tech Law Journal. @:

<http://www.techlawjournal.com/atr/20000914posner.asp>

It was mentioned above that the open structure of the Internet has facilitated the easy access to the market place for minor players, thus bringing equality in competition. A weakness in this context is easily observed as the problem of co-ordinating a project seems to rise quite inevitably if the co-developers are scattered around the world. Another problem with the commons model is that it does not yield the creator any direct financial benefits. Weiser points out that Sun Microsystem's experiment with their open source Java standard has actually earned more profits for Sun's competitors than the company itself. In fact it is questioned if the investment Sun has made in developing Java was worthwhile in the first place.[60]

The propriety control model is not faultless, either. It has been empirically noted that small companies actually tend to be more efficient in their innovation than large monopolies. Even if it is conceded that the large size of an enterprise can be beneficial in traditional industries, it seems to be less true in the information industries where intellectual capital has taken over from physical capital. Weiser points out the shortcoming in Schumpeterian theory of temporary monopolies using Microsoft as an example: it might be that Microsoft is replaced by a rival sometime in the future but in the meantime it still can exercise its monopoly powers to the detriment of rivals and consumers by utilising its de facto industry standards. A particular example is how Microsoft has undermined support for Sun's Java standard. At a fundamental level, it has been shown in several studies that the incumbent monopolist fails to innovate and develop new technologies and sometimes even does the opposite by thwarting the attempts of its rivals to introduce more advanced products.[61]

e. Copyleft and conceptual misnomers

We have looked at the properties of both commons and proprietary control models and weighed the strengths and weaknesses of each. From the legal point of view the commons model in this context is a new phenomenon and therefore in a state of constant flux. That warrants a closer look into the terminology that is being used by the commons theory supporters together with an examination of the concept of copyleft.

The description of copyleft is problematic. On the web site of the Free Software Foundation (FSF) it is stated that the simplest way to make a computer programme free is to put in the public domain, uncopyrighted.[62] In stating this, the FSF fails to pay heed to the way the concept of copyright is formulated. Unlike patents or trademarks, copyright is not a right that needs to be registered in order to gain protection against infringements. In the United Kingdom copyright protection is conferred on the expression of a literary work, such as a computer programme, the moment it is created and fixed in some way, e.g. published on the Internet.[63] It can be argued that a work cannot be uncopyrighted like FSF claims. A computer programme can fail the originality test and lack copyright from the start or it can lose the

---

[60] Weiser, Internet, 579

[61] Id, 580-583

[62] See @: <http://www.gnu.org/licenses/licenses.html#WhatIsCopyleft>

[63] For the definition of copyright in the United Kingdom see The UK Patent Office @: <http://www.patent.gov.uk/copy/definition.htm> and in the United States the U.S. Copyright Office @: <http://www.copyright.gov/circs/circ1.html#wci>

copyright protection due to the period of protection running out, but it cannot it seems, be uncopyrighted as a positive act by the creator or a third party.

Copyleft therefore retains copyright, but by using the GNU-GPL the creator of a program also grants everyone the right to use, modify and distribute the programme on the condition that the licensee also grants similar rights over the modifications he or she makes. Two remarkable copyleft success stories are GNU-Linux operating system and Apache server software. Whilst Stallman first created his GPL and copyleft because of ideological reasons, there is far too large a body of programmers working on copylefted projects today to explain the success on ideology alone. In the year 2000, 63 per cent of web servers were using Apache and Linux was gaining ground on Microsoft in the market share of server operating systems.[64]

It seems that despite the ideal of Free Software Foundation of a society where intellectual property rights are made redundant or diminished in importance in the very least, there remains a strong case for intellectual property rights within the Open Source movement. The inherent danger and shortcoming with open-sourcing the code is the free-riding involved: some developers might take an open source code, duly modify and improve it and use it as a part of their commercial software package thus undermining the freedom of other developers as far as that particular piece of software is concerned. To combat this phenomenon control is needed. This control is currently achieved by licensing the intellectual property rights to the software. The original example is Stallman's GNU-GPL, which has been noted as an aggressive approach to both copyright and contract law. [65]

Therefore it is obvious that "open" in open source is actually quite strictly controlled, although the ultimate goal of control in this context is to reach a level of greater access to source code. Wagner reminds that keeping in mind the above facts it is rather peculiar that eminent supporters of open source software like Lessig and Stallman would condemn those same tools of intellectual property law that that actually provide them with the means they need in reaching their ideal.[66]

III. Is There a Future?

a. Will open source survive the legal attack?

We have asserted above that the open source software industry depends on intellectual property rights. When the open source community uses licences to control their works, they employ the means of contract law and copyright law to protect the openness of their code. At the time of writing this essay, the validity of open source licences is yet to be tested in a court of law despite their relatively long existence. In traditional socio-legal analysis this has been credited to the special nature of the open source community, their internal rules that effectively discourage any breach of an

---

[64] M Mustonen, "Copyleft – the economics of Linux and other open source software" (2003) 15 *Information Economics and Policy* 99, 102

[65] Wagner, Information, 1030

[66] Id, 1031

OSS licence.[67] This, however, is not the whole truth of the matter as several commentators agree that the existence of the GPL is primarily justified because it is a robust legal tool and not a mere social agreement.[68]

During the course of 2003, the entire existence of open source based software has come under threat following a well-publicised band of lawsuits in the United States. Some proponents of open source have expressed their wish in the past that validity of the GPL on which the Linux operating system is based, should be tested in the courts.[69] It seems that they finally have their wish in the form of an action filed by IBM. The fact that a cross-industry giant like IBM is teaming up with the open source community is a noteworthy one. Apparently the Open Source Initiative of realising the goal it set in its early days when making open source software attractive for the corporate users was agreed as the main target of the movement.[70]

It has not been universally accepted, though, that testing the GPL in litigation would be a good idea. O'Sullivan says that a court case, even if the outcome is eventually successful to the open source community, might undermine the comradeship of hackers and turn their self sufficient community into a one depending on extraneous legal processes over which they do not have any control. Therefore, it is questioned whether litigation would be too big a risk and asserted that it might be better to seek to codify the GPL before going to court.[71]

b. MySQL: The first court appearance

The currently pending action between SCO and IBM (which we will look in detail below) notwithstanding, the closest thing to an authoritative opinion from the bench regarding the open source world is the settled action between the United States based Nusphere Corporation and a small Swedish open source based software company MySQL.[72] This case originated when Nusphere as plaintiffs filed the suit in a Federal Court in Massachusetts claiming breach of contract, tortious interference with third-party contracts and relationships, and unfair competition among other business related torts on 15 June, 2001. MySQL duly filed a counterclaim in which they alleged

---

[67] M O'Sullivan, "Making copyright ambidextrous: an expose of copyleft" *The Journal of Information, Law and Technology (JILT)* 2002 (3), ch 3.5 discussing the socio-legal analysis (hereafter referred to as O'Sullivan, Ambidextrous)

[68] See, for example D McGowan, "Legal implications of open source software" 2001 *University of Illinois Law Review* 241, 287 where the author commends Richard Stallman for his "elegant use of the existing property rights structure" in constructing the GPL system. Professor Moglen counters the view of a social contract in a recent interview where he asserts that the GPL is a carefully crafted piece of work to maximise the opportunities that copyright holders can enjoy under the Berne Convention on copyright. See A Orlowski, "The GPL will win, claims law prof." 19 Aug 2003 *The Register*. @: <http://www.theregister.co.uk/content/4/32393.html>

[69] E Moglen, "Free software matters: enforcing the GPL II" 10 Sep 2001 *Linux User* available @:

<http://moglen.law.columbia.edu/publications/lu-13.html>

[70] See note 12 above

[71] O'Sullivan, Ambidextrous, ch 3.6 discussing Moglen's enthusiasm of proving the GPL in litigation

[72] MySQL AB and Nusphere Corporation announced settlement, see 7 Nov 2002 Company Press Release @: <http://www.mysql.com/press/release_2002_14.html> and L A Majerus, "Court evaluates meaning of 'derivative work' in open source license". @: <http://articles.corporate.findlaw.com/articles/file/00050/008924>

trademark infringement, breach of the interim agreement between the parties and breach of the GNU-GPL.

In the interim agreement between the parties it had been stipulated that software created by MySQL would be published under the GPL. Keeping in mind the viral effect of the GPL it would initially seem that Nusphere had failed to respect the conditions under the licence when it released a commercial software bundle called Gemini that included MySQL's code, without releasing Gemini's source code. In the later releases, whilst the case was still pending in court, Nusphere did include the source code for Gemini in their bundle. Sticking to the GPL clauses, MySQL claimed that Gemini was a derivative work in the sense purported in the GPL.

Professor Moglen submitted an affidavit in support of the defendants/counter-plaintiffs stating the three primary conditions of distributing GPL programs: 1) No additional licence conditions: Redistribution must itself occur under the GPL and only the GPL, with no additional licence conditions. 2) Source code: Redistribution must include "source code", the human-form of computer programmes that allows programmers to understand and modify computer programmes for themselves. 3) Inclusion of GPL: Redistribution must include a copy of the GPL, so that users are aware of their rights to use, copy, modify and distribute, and so that anyone engaged in redistribution is also aware of the conditions under which redistribution is permitted.[73]

MySQL did indeed win the initial favour of the court in a preliminary injunction on February 28, 2002:[74]

> *MySQL has not demonstrated a substantial likelihood of success on the merits or irreparable harm. Affidavits submitted by the parties' experts raise a factual dispute concerning whether the Gemini program is a derivative or an independent and separate work under GPL, para 2. After hearing, MySQL seems to have the better argument here, but the matter is one of fair dispute. Moreover, I am not persuaded that the release of the Gemini source code in July 2001 didn't cure the breach.*

Judge Saris did not rule on MySQL's request for a summary judgment of the breach of contract under the GPL but gave the parties a strong hint that the defendants/counter-plaintiffs had a strong case, "…MySQL seems to have the better argument here…" thus effectively facilitating the settlement between the parties. The parties' behaviour seems to strongly support the commentators who have previously concluded that the GPL is not only a "social contract" but an enforceable legal tool: yet another case was settled out of court with a favourable outcome to the open source community.

c. SCO, IBM, and Linux: The current case

SCO (previously known as Caldera) is a Utah based company that owns the intellectual property rights to Unix operating system.[75] They make money by

---

[73] E Moglen, "Declaration of Eben Moglen in support of defendant's motion for a preliminary injunction on its counterclaims", para 19. @:< http://www.gnu.org/press/mysql-affidavit.pdf>

[74] 195 F. Supp. 2d 328 (D. Mass)

licensing the rights to Unix to companies that make their own versions of Unix, one of those companies being IBM with their AIX version of Unix. In March 2003 SCO filed a lawsuit against IBM alleging that the latter had improperly taken confidential information included in Unix code and contributed it to Linux, which is an open source based operating system. SCO originally sought $1 billion in damages but amended the claim to $3 billion in June 2003.[76] In May 2003 SCO approached some 1350 corporations using Linux with a letter containing a warning that Linux is an unauthorised derivative of Unix to which they own intellectual property rights and that legal liability for the use of Linux may extend to commercial users. At the time of writing, the latest move in "Linux wars" as the phenomenon has been dubbed is IBM's counterclaim filed in August 2003. IBM's counterclaim is of special interest because it alleges breach of the GPL on SCO's behalf and could finally lead into an authoritative answer to the question of GPL's enforceability.

1. Are Linux users infringing SCO's copyright?

The above mentioned letter by SCO to their customers suggesting potential legal action should they not buy a licence to use the Linux operating system has been removed from SCO's web site after IBM and Red Hat[77] filed lawsuits against SCO. The manner of SCO's conduct toward the clientele while the lawsuits are still in their early stages illustrates a point that clearly appears to tip the balance of probable outcome in the favour of the open source community. The initial letter to SCO's customers left much to be desired in terms of legal clarity, being most ambiguous in addressing the type of infringement it alleged.[78] SCO did however, on a later date elaborate their claims and asserted that a copyright infringement had taken place as a result of certain versions of the Linux operating system containing code inappropriately copied from SCO's proprietary Unix code.[79] Despite the fact that several commentators have agreed that the plaintiff's havering stems from the fact that its case is a weak one, it has to be kept in mind that impartial analysis of this case was not available at the time of writing.[80]

---

[75] Unix was originally developed by AT&T at Bell Labs, the rights were sold to Novell in 1993 who subsequently sold them to SCO in 1995. For the history of Unix, see Lucent Technologies web site @: <http://www.bell-labs.com/history/unix/>

[76] *The SCO Group, Inc. vs. International Business Machines Corporation (pending)*

In the United States District Court, District of Utah

Case No. 03-CV-0294, filed in June 2003. Text of the complaint is available on plaintiff's web site @: <http://www.sco.com/ibmlawsuit/amendedcomplaintjune16.html>

[77] Red Hat, "Red Hat Takes Aim at Infringement Claims" 4 Aug 2003 press release

[78] E Moglen, "Questioning SCO: a hard look at nebulous claims". This Open Source Development Lab's position paper is available @: <http://www.osdl.org/docs/osdl_eben_moglen_position_paper.pdf> (hereafter referred to as Moglen, Nebulous)

[79] SCO, "SCO registers UNIX® copyrights and offers UNIX license" 21 July 2003 press release. @: <http://ir.sco.com/ReleaseDetail.cfm?ReleaseID=114170>

[80] Albeit the majority of commentators I have come across seem to favour the defendants, there are opinions voiced in support of the plaintiffs as well. For an alternative opinion see R Enderle, "SCO vs. IBM: the other reality" 2 Sep 2003 *Linux Insider*. @: <http://www.linuxinsider.com/perl/story/31479.html>

Let us first examine a situation where a lawfully acquired copy of the Linux operating system is being used merely for a company's internal work. Every time a computer programme is loaded from a mass storage device, e.g. hard disk or CD into the Random Access Memory (RAM), a new copy is arguably made. The acts restricted only to the holder of the copyright concerning users of computer software in the European Union can be read in Article 4 of the Software Directive. These include permanent or temporary reproduction, making an adaptation (derivative work), and distribution.[81] The strict wording of Article 4 (a) that the right holder has the exclusive right to do or authorise

> *[T]he permanent or temporary reproduction of a computer program by any means and in any form, in part or in whole. Insofar as loading, displaying, running, transmission or storage of the computer program necessitate such reproduction, such acts shall be subject to authorisation by the rightholder*

has lead to misinterpretations in some instances when the commentator has failed to read the Directive as whole.[82] One has to keep in mind that Article 4 is qualified being

> *Subject to the provisions of Articles 5 and 6" and in Article 5 it is stated that "In the absence of specific contractual provisions, the acts referred to in Article 4 (a) and (b) shall not require authorisation by the rightholder where they are necessary for the use of the computer program by the lawful acquirer in accordance with its intended purpose, including for error correction.*

In the United States, the seat of the current litigation, there is a similar provision in the US Copyright Act, section 117 regarding the limitation of exclusive rights in computer programs.[83]

> *(a) Notwithstanding the provisions of section 106 it is not an infringement for the owner of a copy of a computer program to make or authorise the making of another copy or adaptation of that computer program provided:*
>
> > *(1) that such a new copy or adaptation is created as an essential step in the utilisation of the computer program in conjunction with a machine and that it is used in no other manner…*

Thus Article 5 in the EU and section 117 in the US assure that a "lawful acquirer" (EU) or "owner" (US) is entitled to use their software without any further licence

---

[81] See Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs OJ 1991 L 122, which was implemented in the United Kingdom by Statutory Instrument 3233/92, amending the Copyright, Designs, and Patents Act (1988).

[82] A Orlowski quoting G Lea in, "Against SCO's GPL jihad: one size doesn't fit all" 25 Aug 2003 *The Register* @: <http://www.theregister.co.uk/content/35/32479.html>. In the article British law lecturer Lea is reportedly asserting that the situation regarding the user's right to make a copy of a computer program in order to secure its functionality in the EU differs from that in the US which is not the case, as will be shown below

[83] 17 U.S.C. Section 117 available on-line @: <http://www.copyright.gov/title17/92chap1.html#117>

from the owner of the copyright even though their usage involves the inevitable making of a copy in the RAM memory of the computer.

Despite the above analysis, there remains at least in theory, a possibility of the companies using Linux facing an infringement claim. It all seems to depend on the pending breach of contract suit between SCO and IBM.[84] If SCO is to prevail in the suit against IBM and manages to establish that the code included in Linux indeed improperly incorporated confidential material, then the lack of any indemnity clauses in the GPL could subject the companies to damages as well. However, there are some considerations to keep in mind that will mitigate the alleged breach of Linux users on the one hand, and the likelihood of their being subjected to damages, on the other. First, it should be remembered that confidential information can lose its trade secret status when it becomes public knowledge through no fault of the recipient.[85] The progress is rapid in the context of computer software and it is fairly feasible to assume that a piece of code that has been a trade secret two years earlier, might have already lost its status on the day a company bought their Linux package. Second, as Rosen points out, it is possible that a court does not allow damages for the second time on the same infringement. He asserts that IBM is acting as a shield for the Linux users.[86]

Consequently, it seems that companies using Linux are not in immediate danger of infringement, although a theoretical possibility exists as long as the case between SCO and IBM is pending. Whilst the end users of the Linux kernel appear to be relatively well protected from claims, this is not necessarily the case with IBM which has contractual obligations with the plaintiff. What follows is a discussion of currently pending litigation that promises to clarify the status of copylefted software.

2. Validity of the GPL: SCO vs. IBM litigation

The string of events that now seems inevitably to lead into testing the validity of the GPL in court, started with SCO filing the lawsuit against IBM in March 2003. Both suits appear to hold quite a bit of interest regarding the future viability of the open source community as the GPL features prominently in each one.

In the original case, SCO is claiming breach of contract[87], unfair competition[88] and misappropriation of trade secrets[89]. IBM had entered into a licensing contract with the original developers of Unix, AT&T Technologies, whose successor plaintiff is SCO. The contract provided that IBM was allowed to use licensed code in its own variations of Unix but contained strict limitations on use and distribution to protect the

---

[84] See note 87 below

[85] Utah Code, Title 13, Chapter 24, Section 2 on trade secrets, available on-line @: <http://www.le.state.ut.us/~code/TITLE13/htm/13_18003.htm>. Trade secrets in the United States are not regulated on federal level but most states have adopted Uniform Trade Secrets Act, including Utah where the plaintiff is incorporated

[86] See Lawrence Rosen's interview in P Rooney, "Linux customers, partners can ignore SCO-IBM-Red Hat Linux Battle, OSDL Says" 14 Aug 2003 *CRN News*. @: <http://www.crn.com/sections/BreakingNews/dailyarchives.asp?ArticleID=43929>

[87] See the amended complaint in *The SCO Group, Inc. vs. International Business Machines Corporation (pending)*, para 103, 127, and 133.

[88] Id, para 147

[89] Id, para 160

confidential and proprietary property.[90] The dispute originated when SCO claimed that IBM had actively supported, assisted and promoted the transfer of certain components of said confidential code into the Linux operating system thus undermining the commercial value of SCO's proprietary code.[91]

The GPL lies very much at the heart of this dispute. Linux is made up of copyrighted contributions of thousands of programmers around the world. Their combined effort is then published and distributed under the GPL, which gives everyone permission to copy, modify and distribute the code as long as all this occurs under the GPL. Owing to the GPL's terms, all these pieces of code that together comprise the kernel of the Linux operating system are deemed to be open source.

It has been asserted that the fundamental problem with the plaintiff's claim of improper distribution of proprietary code is the fact that the plaintiff itself has distributed the same version of the Linux kernel that is the object of current dispute, under the GPL in the capacity of its business as a distributor of the Linux operating system.[92] Because of the terms of the GPL, the plaintiff does not appear to be in a position to claim that it has an exclusive right to distribute Linux, excluding the defendant and various other distributors. The wording of Section 6 of the GPL provides:

> *Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights herein.*[93]

It seems clear from the wording of the text that once anyone releases a copy of a programme containing GPL'd code, they agree it to be freely copied, distributed and modified in the spirit of copyleft. It would appear to be an unwise move from the plaintiff in the instant case to embark on litigation with the facts speaking against them. Several commentators have pointed out this fact, predicting a successful outcome to the defendant.[94] The plaintiff's case is, however, based on the assertion that SCO never authorised the release of certain pieces of their Unix code to be included in the Linux kernel, and that the defendant has used confidential information to the detriment of the plaintiff by releasing the code owned by the plaintiff under the GPL. This is where the parties' opinions of relevant facts differ and the court will ultimately have to decide which one is right. Attorney Mark Heise, representing the plaintiff says, "The difference [between SCO and other contributors to the Linux kernel] is that other companies have donated their copyrighted material, and they did

---

[90] Id, para 60

[91] Id, para 106. IBM is one of the major hardware suppliers who have looked into open source technology especially in server software. Using Linux instead of proprietary products such as those produced by Microsoft has enabled IBM to offer more attractive deals with customers who are no longer dependent on a single supplier's software license conditions

[92] See C Zymaris, "Why SCO's attack on the GPL will fail" 21 Aug 2003 *Sydney Morning Herald*. @: <www.smh.com.au/articles/2003/08/21/1061434974481.html>

[93] GNU General Public License, Version 2, June 1991 @: <http://www.gnu.org/licenses/gpl.txt>

[94] See note 80 above. The fact that plaintiff still reportedly continues to distribute Linux kernel on their FTP site is an interesting one and can be seen to weaken their case. See note 92 above

so knowingly, and they are free to do that. But you are not free to take somebody else's copyrighted or otherwise protected material and put it into the GPL and suddenly it is for everybody".[95] Professor Moglen, speaking for the open source community counters his opinion: "SCO, as Caldera, has indeed contributed to the Linux kernel, and its contributions are included in modules containing GPL notices."[96]

The element of ambiguity involved in SCO's contribution to the Linux kernel is possibly the reason why the plaintiff SCO has announced that it contests the validity of the GPL itself. Should this practical embodiment of copyleft ideology be annulled, the impact could be fatal to the open source community with commercial enterprises potentially losing the faith they have started to show in it. In the short run, proprietary software producing giants like Microsoft would undoubtedly win if the sales of open source based products plummeted due to uncertainty of their legal status.

At the time of writing in September 2003 the litigation is only at its infancy thus rendering it extremely difficult to analyse the possible outcome, keeping it in mind that complex intellectual property related cases tend to take years to reach a conclusion. The plaintiff SCO has, however, publicly vowed to challenge the validity of the GPL – and since their legal representative has elaborated the grounds of this assertion, we are already in the position to examine whether or not their arguments appear to have merit.

In an interview with a CNET News.com reporter in August 2003, attorney Mark Heise reportedly asserted:

> *We don't think the GPL applies. We believe it is pre-empted by the federal copyright law … If you look at the terms of the GPL and the terms of copyright law, copyright law governs. It is the exclusive authority regarding the use, distribution, etc., of copyrighted material. In the GPL, (there is a section that) specifically says it applies only to the use and distribution. In other words, the exact same topics that are covered exclusively by the Copyright Act are covered by the GPL. Section 301 of the Copyright Act says the Copyright act pre-empts any claims that are governed regarding use, distribution and copying. We believe that although the GPL is being tossed into the fray, it is pre-emptied by federal copyright law.*[97]

Legislative history shows that *ratio legis* behind enacting section 301 of the US Copyright Act was the desire to accomplish a single federal system of copyright.[98] The US system of copyright was marred by dualistic application of common law copyright which encompassed unpublished works and statutory copyright regulating published works. In order to promote national uniformity, it was declared that, "The

---

[95] See L M Bowman, "SCO's big legal gun takes aim" 21 Aug 2003 *CNET News*. @: <http://news.com.com/2008-1082-5066520.html> (hereafter referred to as Bowman, Big gun)

[96] See Moglen, Nebulous

[97] See Bowman, Big gun

[98] See Historical and Revision Notes regarding Section 301, House Report No. 94-1476 @: <http://uscode.house.gov/DOWNLOAD/17C3.DOC>

intention of section 301 is to preempt and abolish any rights under the common law or statutes of a State that are equivalent to copyright and that extend to works coming within the scope of the Federal copyright law." It was further stated that the declaration of the principle in section 301 was intended to be written in the clearest and most unequivocal language possible.[99]

The wording of section 301 (a) is very clear:

> *On and after January 1, 1978, all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106 in works of authorship that are fixed in a tangible medium of expression and come within the subject matter of copyright as specified by sections 102 and 103, whether created before or after that date and whether published or unpublished, are governed exclusively by this title. Thereafter, no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State.*[100]

Section 301 refers to section 106 of the US Copyright Act which lays out the five fundamental rights that have been exclusively reserved to copyright owners: reproduction, adaptation, publication, performance, and display.[101] The first three are relevant to the instant litigation:

> *The owner of copyright under this title has the exclusive rights to do and to authorize any of the following: (1) to reproduce the copyrighted work in copies or phonorecords; (2) to prepare derivative works based upon the copyrighted work; (3) to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending.*[102]

In the light of the legislator's will, there is no doubt that the reason for enacting section 301 of the US Copyright Act was to simplify the domestic system of copyright in the US that had been deemed too complicated. There is no suggestion that this pre-emption should encompass a means of authorising a third party to enjoy the rights enumerated in section 106, i.e. a licence agreement in case of computer software. To my understanding such interpretation would render any copyright related licence dealing with the five fundamental rights invalid. In fact, it appears that the system of sections 301 and 106 facilitates the foundations of licensing by enabling the copyright holder to authorise a third party to take advantage of the copyright holder's right.

Based on this analysis it seems that the plaintiff's arguments purporting to invalidate the GPL are not particularly strong at this time, but as it bears repeating, the litigation is still in early days at the time of writing.

---

[99] Id, para 4

[100] See 17 U.S.C. section 301, available on-line @: <http://www.copyright.gov/title17/92chap1.html#301>

[101] See Historical and Revision Notes regarding Section 106, House Report No. 94-1476 at <http://uscode.house.gov/DOWNLOAD/17C1.DOC>

[102] See 17 U.S.C. section 106, available on-line @: <http://www.copyright.gov/title17/92chap1.html#106>

3. IBM's counterclaims: SCO is in violation of the GPL

IBM filed a counter suit against SCO in August 2003.[103] In addition to four counts of alleged patent infringements and other business related claims, they claim that SCO has breached the GPL, asserting that SCO accepted the terms of the GPL by itself distributing the version of Linux kernel that included the pieces of code SCO is claiming to be trade secrets to the public under the GPL.[104] According to the counter claim, SCO has nevertheless claimed ownership rights over Linux code that includes IBM contributions; sought to collect licence fees with respect to Linux code; copied, modified, and distributed Linux on terms other than those set out in the GPL; and sought to impose additional restrictions on the recipients of the Linux code.[105]

One has to weigh this assertion against the fact that SCO based its original claim on confidential code being improperly included in Linux but the irreversible fact remains that SCO itself was selling that same version, and financially benefiting from the sales of the very same pieces of code under the GPL, that it claims to be a trade secret. This appears to speak for IBM, since voluntarily releasing material under an open source licence does not support the fact that it represents a noteworthy financial value as a trade secret.[106]

IV. Conclusions

In this essay I have purported to examine the roots of Open Source Software movement, to find an explanation for the motivation behind skilled professionals working in an industry that does not appear to function according to the rules of the traditional economic model, and finally discuss whether or not open source can continue to be an option for businesses in the light of recent litigation questioning the validity of copyleft in the United States.

It is undoubtedly correct to assert that there are several benefits in using open source software in a commercial context.[107] It can guarantee higher reliability and quality in software, and promotes open standards that can result in rapid growth – the Internet itself through its essential open sourced TCP/IP protocol being the most obvious example. Open source frees end users from a single supplier controlled market and is very future-proof because a huge pool of independent developers are working on successful projects, assuring the existence of necessary updates.

---

[103] See IBM, "Defendant IBM's Answer to The Amended Complaint and Counterclaim-Plaintiff IBM's Claims Against SCO" 6 August 2003. At the time of writing (September 2003) defendant's answer was not yet available on Utah District Court's web site where the case docket can be accessed @: <http://www.utd.uscourts.gov/documents/ibm_hist.html> but instead could be found @: <http://lwn.net/Articles/43529/>

[104] Id, para 76

[105] Id, para 78

[106] For a definition of trade secret in Utah, see Utah Code, Title 13, Chapter 24, on trade secrets, available on-line @: <http://www.le.state.ut.us/~code/TITLE13/htm/13_18003.htm>

[107] See, for example E S Raymond, "The Magic Cauldron", ch 10.1. @: <http://catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/>

For a commercial firm there are, however, some serious legal caveats to keep in mind.[108] The obvious concern in the light of the topic of this paper is the fact that a company which completely abides by their GPL responsibilities, for example by merely using separately purchased Linux operating systems in their internal work, can still be a subject of an infringement claim if there has been a violation by an upstream contributor, like allegedly IBM. The lack of an indemnity clause in the GPL leaves downstream end-users at the mercy of all contributors' good faith.

Another concern is internal management in a software producing company. One needs little imagination in drawing up a situation where an individual programmer working for a large company, for example SCO, that distributes and (allegedly in SCO's case) contributes to the Linux kernel, unbeknownst to the company's senior staff incorporates a piece of code that actually is worth money as a trade secret in a modified version of Linux and this version is released under the GPL before the mistake is noticed. In short, using open source solutions in a company's strategy requires careful planning and must be controlled centrally within the company.

Despite these well founded concerns I believe that open source software can co-exist with proprietary programs. In some instances it can be more plausible to choose open code than keep it closed: ultimately it is up to the firm's business strategy. In any case, the ideology of commons in the field of software is here to stay, and legislation, as well as commercial actors will adapt to it like they always have when a new way of doing business has arrived.

---

[108] For a general discussion, see R Raysman and P Brown, "Using Open Source Code in Proprietary Products" (2001) 226 New York Law Journal 112