

Mikael Pawlo*

Efficiency, Innovation, and Transparency – The Future of Intellectual Property Rights

– *Why are they after me?*

In the movie *Antitrust*, Tim Robbins, with his usual excellence, plays the part of the Bill Gates character. When the Robbins character blurts out his desperation it is because the US Department of Justice is on his tail, exploring the innermost secret of the code in Robbins' computer programs. In one of the crucial scenes where Robbins' character eventually loses control over his code, Robbins still cannot understand why his protégé Ryan Phillippe's character is working against him. After all, the code is mine, Robbins' character concludes. Should not Robbins as the copyright proprietor be able to decide just what to do with his computer programs? Should not the legislator protect the Robbinses of our world from the efforts of self-appointed Phillippe freedom fighters to release and reveal the Robbins code to the world? Only to a certain point.

1 Experimental copyright in action

The number one full-scale experiment on intellectual property in history is now in practice. I am referring to the new types of licenses for computer programs: free software and open source. We are looking at an experiment that will define the future of intellectual property.

Free software, as defined by Richard M Stallman, rests on four foundations:

- You are free to run the program, for any purpose.
- You are free to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)
- You are free to redistribute copies, either gratis or for a fee.
- You are free to distribute modified versions of the program, so that the community can benefit from your improvements.

Free software is very simple in its construction. It uses the provisions of copyright law whereby the author has an exclusive economic right in his work. In copyright law, computer programs are regarded as literary works. Thus, the author of a computer program can enter into any agreement regarding his work. One such agreement is the GNU GPL. GNU GPL stands for GNU General Public License, while GNU is a "recursive" abbreviation of Gnu's Not Unix. GNU is the manifestation in practice of free software and Richard M Stallman's

* *Mikael Pawlo* is an associate with the Stockholm office of the law firm Lindahl where he specialises in IT law, commercial law, and intellectual property law. Among other things, he has been chairman of the Swedish professional organisation of on-line vendors of contents and services (BitoS). He is also the Swedish editor of the Nordic Intellectual Property Review (NIR).

attempt at building a free Unix system. The most famous part of the GNU system is the kernel developed by Linus Torvalds under the name Linux. The GNU GPL that lays the foundation of free software is enforceable both under the principle of freedom of contract and through copyright law. According to Stallman's legal counsel, Professor Eben, the GNU GPL has yet to be successfully challenged. As I write this, in the spring of 2002, in a decision handed down in Boston, US District Judge Patti B. Saris has ruled on the preliminary injunction motion in *MySQL AB vs. Progress Software Corp.* That case is often referred to as the first test in court of the GNU GPL. It is a complicated case with several components. In the matter of Progress's distribution rights under GNU GPL, Saris did not grant an injunction. In the public hearing, Judge Saris made clear that she sees the GNU GPL as an enforceable and binding license, but that as long as Progress Software appears to be presently in compliance with the GNU GPL, there is probably no irreparable harm being caused to MySQL AB, and therefore no case for a preliminary injunction.

Open source is different from free software. Open source is based on a definition designed by Eric S Raymond and Bruce Perens. The basic idea behind open source is simple: when programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, and people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing. Raymond and Perens designed the open source definition. Open source is less restrictive than GNU GPL and free software, but it does not just mean access to the source code. Open source is not a license, but a set of rules that any license claiming to be open source must follow. The most important clause in the open source definition requires the distribution terms of open-source software to comply with the following criteria:

“The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost – preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed”.

The Open Source Definition is described as a bill of rights for the computer user. It is not a developed philosophy like free software, but maintains a more pragmatic hands-on approach.

It is often said that Rome gave civilisation the law. That may be true, but someone else invented intellectual property law. According to Stewart – an acclaimed scholar on international copyright law – the early Greeks and Romans had a developed notion of authorship, which was confined to the desire of teachers and philosophers to be credited for their own teachings. This was a moral question, thus not regulated in law.

Most people agree that the first copyright law was the English Statute of Anne passed in 1709. The system used today in most Western societies derives from the Berne Convention of 1886. Some things have changed over time, but only in favour of stronger protection of the author and the copyright holder. The one common principle is simple and almost globally applicable: with few exceptions, you need the copyright holder's permission if you want to make new copies or create a work deriving from the author's work within seventy years of the author's death.

2 Freedom of speech challenged

The Romans took a broad view of contract law and other essentials of civil law. Details may vary over time and between jurisdictions, but there is little controversy about the basics. Copyright, however, is widely debated these days. American scholars Lawrence Lessig, Jessica Litman and Siva Vaidhyanathan produced the most famous recent works in the area, following a long European tradition of debating the author's rights. You may think that the time for copyright protection – life plus seventy – is too long. You may think that fair use is too limited. You may think that the Russian programmer Dmitry Sklyarov should never have been prosecuted under the DMCA (the Digital Millennium Copyright Act) for designing an anti-circumvention device for e-books. You may think all these things, and Lessig, Litman and Vaidhyanathan very eloquently put them all, but I think the issue of copyright protection of computer programs – of code – is different in principle. In his book “Code and other laws of cyberspace” Lessig has demonstrated that code, i.e. programmed functions of computer systems, can be more important than law.¹ Computer programs should never have been protected as literary works in the first place. That just happened. But now that it is time for a change, I think the great experiment that we are all taking part in is a wonderful way – through freedom of contract – to experiment towards a new legal take on code.

Free software and open source could together be described as open code. With open code, I mean that the source code is available to the user and the development of the computer program is decentralised. It is often argued from the experience of Linux, Apache and Sendmail that the distributed development process of open code is good for security, speed of development and interoperability.

Lessig argues in his book “Code” that code could be more important than law, when it comes to free speech in computer networks. Lessig concurs that we should think about the architecture of cyberspace – its “code” – as a kind of regulator; that this regulator is likely to regulate more than law does today; that “doing nothing” is to lose some of the freedom the Internet now guarantees. The code – by not being transparent – may threaten freedom of speech. What if the code in itself makes certain types of expression void? Freedom of speech would then be stifled through the architecture of the online, Internet or IT environment. And this could happen without any political debate.

Furthermore, open code is good for consumer and customer confidence and trust. Would you trust a product that you are not allowed to disassemble? What if the product carried all your personal data? The trust and transparency argument is in my opinion the strongest argument for open code legislation.

3 Open code legislation

One of the big issues of free software during 2001 was whether Richard M Stallman was for or against a codified GNU GPL. Hence, did Stallman – the father of free software – propagate a law to support his beliefs?

Tim O'Reilly tried to press the issue in a couple of articles and seemed convinced that Stallman and his colleague Bradley M Kuhn were for GNU GPL legislation. O'Reilly suggested a system where developers themselves choose the rules under which they release software, not very much different from the system in effect today. Eric S Raymond wrote a satire to prove how wrong Stallman and Kuhn would be to suggest a GNU GPL law. Raymond posed Stallman and Kuhn the question whether they would get a law passed making proprietary licenses illegal if they could. Stallman and Kuhn leaned slightly towards the legislative point of view, but never gave a straight answer whether they were for or against a

¹ Lawrence Lessig, *Code: And Other Laws Of Cyberspace*, New York: Basic Books 1999.

codified GNU GPL. Stallman and Kuhn wrote: “We believe, though, that with time, as more and more users realize that code is law, and come to feel that they too deserve freedom, they will see the importance of the freedoms we stand for – just as more and more users have come to appreciate the practical value of the free software we have developed.”

As stated above, copyright law is often questioned. In an article in Wired 1994, John Perry Barlow wrote that copyright was not designed to protect ideas or bits of information but only to protect ideas as expressed in fixed form. Hence, according to Barlow copyright is dead in the digital age.

Copyright was made to create an incentive for authors and scientists to create and explore and give them a guarantee that they would profit from their creations. A copyright system that is too strict in favour of the authors will work as a hinder and not an incentive for creativity. In the epilogue of his book *Copyrights and copywrongs* Siva Vaidhyanathan states that “a looser copyright system would produce more James Bond books, not fewer. Some might be excellent. Other might be crappy. Publishers and readers could sort out the difference for themselves. The law need not to skew the balance as it has.”²

4 “Lagom” copyright for computer programs

In Sweden we have one word that I have yet to find anywhere else. The word is “lagom” and it defines the space between too much and too little. Lagom could be translated into “moderate” or “just right”, it is the situation where the glass is not half-full or half-empty – it is lagom filled. We need “lagom” copyright for computer programs because computer programs are written incrementally. That means that it is important to be able to reuse previously written code. Hence, you need to be able to write the computer program without the original author being present in your project. The aforesaid is a strong argument for a codified GNU GPL, since one of the cornerstones of GNU GPL is the right to reuse previously written code. Further, examination of the code is important for interoperability. Interoperability means that computer programs should contain interchangeability, one should be able to substitute one computer program for another, and connectability, that is the ability of one computer program to function with another.

The European debate on interoperability ended in 1991, when the European Union introduced a directive on the Legal Protection of Computer Programs. The directive exempts ideas underlying any element of a computer program, including its interfaces, from copyright protection. It also specifically permits disassembly of computer programs in order to achieve interoperability. Transparency is therefore ensured, but without access to the source code of the computer program it would still be hard to disassemble and interpret the functions of the computer programs. The GNU GPL wants to solve this by always forcing the developer to disclose and distribute his software.

Would not a modern democratic society benefit from a plurality of irreconcilable and incompatible doctrines? We need the GNU GPL, but we also need proprietary software and open source software. That would make the case for GNU GPL legislation void. However, as Lawrence Lessig concludes in his book *Code*, the code may in itself work against plurality. If we choose to believe Lessig we might want to reconsider regarding computer programs in the same way as literature.

² Siva Vaidhyanathan, *Copyrights and Copywrongs*, New York: New York University Press 2001.

In his book “The Future of Ideas” Lessig suggests a reform of software copyright law forcing computer programmers to disclose their source code when the copyright expires.³ Lessig would protect computer programs for a term of five years, renewable once. Copyright protection would in Lessig’s proposal only be granted if the author put a copy of the source code in escrow. The source code should be disclosed to each and everyone when the copyright expires, perhaps through a server with the U.S. Copyright Office.

That much said, Lessig is very reluctant to make open code a law. In *The Future of Ideas*, Lessig states that the government should “encourage” the development of open code. Such “encouragement” should not be coercive. According to Lessig there is no reason to ban or punish proprietary providers. But this view is hardly consistent with Lessig’s view on the future of software copyright law. In Lessig’s future system proprietary providers are severely punished. They lose about 100 years’ protection, which is life of the author plus seventy years compared to five plus five years and then full disclosure. Lessig’s system is very similar to WIPO’s proposed system of 1970 where copyright protection should be traded for putting the source code in escrow. However, the European development of copyright seems to have been founded on two principles:

1. more copyright (stronger IP laws) is good,
2. everyone should think 1, if only through harmonization.

Lessig’s ideas are not new from a European perspective, but they have revitalized the European copyright debate. In Europe, the debate over the copyright system has not been as intense as the US debate in the recent years. This is probably because the European debate over copyright has been ongoing for the past century and the US debate is quite new. The focus of the European debate on intellectual property development concerns patents on life and software. The European patent system is influenced by the US patent system and more things can be patented in practice than the legislator intended. This creates an interesting situation where the strong European copyright is exported to the US and the strong US patent system is imported, thus creating stronger intellectual property rights in both the US and Europe respectively. The strong US patent was a consequence of the relatively weak copyright protection. Therefore the new legislation creates a situation where the intellectual property protection of computer programs is stronger than ever. But is it good for innovation, and how will it affect the society’s need of transparency?

In an article published in the *Stanford Technology Law Review*, Mathias Strasser argues that any move towards more open code would be highly undesirable from societal point of view, as it would destroy the market-based incentive structure that currently encourages software producers to develop code that consumers find attractive. By applying the utilitarian incentive theory and the Lockean labour-desert theory⁴, Strasser tries to explain why the current copyright system is the best.

Stallman and Moglen have yet to convince me that the GNU GPL and free software philosophy is the final answer to intellectual property protection of computer programs. However, I am not convinced that neither Strasser nor Lessig is right in their view of the software copyright. But I choose to believe Lessig when he states that code is law. The two fundamental principles of European copyright development do not address this issue. The code layer in the networks may in my opinion affect the freedom of speech at large. I do not

³ Lawrence Lessig, *The Future of Ideas*, New York: Random House 2001.

⁴ According to the labour-desert theory, natural resources were given to people by God and title may be lost or abandoned, but anyone might gain title to anything, even resources held in common, if one used labour to convert the natural resources into something useful.

think that copyright is dead in the sense Barlow told us in 1994. Copyright is still around, and even if it's not effective in the digital age – as observed by Barlow – the courts enforce copyright. Therefore, we need to find a new way to deal with copyright protection of computer programs. The U.S. Digital Millennium Copyright Act, the Infosoc EU directive (2001/29/EC) and prohibition on reversed engineering is not the right way to develop copyright. We need more transparency, but still we need to consider the points raised by Mathias Strasser and Tim O'Reilly. It is important that the incentives for larger businesses remain even if the code is more open through a change in the copyright law. If such a change is made, we need to consider the unique characteristics of computer programs. We should not continue to compare computer programs to literary works. Books are not software.

What we need is balance. What we need is “lagom” copyright protection for computer programs. I guess you should take the main parts of the current patent and copyright system and catalyse these systems into the new “lagom” copyright directive. We need to start thinking about these issues soon if we're not aiming to keep our grandchildren stuck with the current system for life.

5 Music and the threat of efficiency

In the past, legislators have designated a private sphere in the life of each individual as unregulated. In your private sphere, you could do many things, as long as they concerned only yourself and maybe some friends. The private sphere was considered your home. You could exercise your fair use rights to copy music and papers for personal or academic use. The Internet tampers with this ancient tradition.

Your means of communication are much more efficient than legislators could have foreseen when the copyright statutes were designed. Making a copy of something for your friends is completely different in the Internet age. You can send the copy to a thousand of your friends with very little effort at a very low cost. It is extremely efficient.

Legislators did not want to regulate the private sphere and did not recognise a need for doing so. Ten years ago, when the Swedish Copyright Act was revised, this was still the position held by the legislators. They were aware of the common practice among friends of copying and distributing mix tapes of favourite songs. Swedish legislators reasoned that it was not a good thing to try to regulate the private sphere, since the legislation would be very hard to enforce. In regulation, one should try to refrain from creating rules that cannot be enforced, since they erode the populace's confidence and trust in the law as something logical and beneficial to society.

But the digitalisation of copyright and the Internet have made it much easier to obtain control over and monitor copyright violation, even if such activities are conducted in the private sphere.

In the mix tape example, there was a physical barrier preventing the communication from reaching efficiency, since distributing the tapes en masse would be prohibitively expensive. When Xerox introduced the copier in 1959, several smaller printing houses were forced to close. In 1966, Xerox introduced the Telecopier (now known as the fax machine). Xerox made copying possible over the physical barrier of distance, but it was still possible to make money on printed works. The improved means of communication and distribution of information represented by the copier and fax machine did not put all journalists and writers out of work, and neither machine was prohibited. Still, it looks like the musical equivalent of these Xerox machines – Napster and its followers – will be prohibited or at least sued out of business. Some intermediaries will die because of the new technology, just like the smaller

printing houses died out when the copier was invented. But is this really an argument for prohibiting technical progress as such?

So, what is the proper balance between the music industry's wishes and the sanctity of your personal sphere? How efficiently will copyright holders and record companies allow us to communicate with each other?

6 Compulsory licensing

For the record, I do not think that music should be free as in free beer. But I do think we need compulsory licensing to stimulate creativity and innovation. Music would then be free as in free speech (but that is another story). It is important that the legislators – and the courts – give users the freedom and the right to a private sphere. Even though enforcement and control of the private sphere could increase with new technology, I do not want record companies and Microsoft to become a private alternative to the Orwellian surveillance state. Stay away from my hard drive. Please. And let me communicate in the most sophisticated and efficient way available, even if it means that you risk losing money from my possible contributory or direct copyright infringement.

To ensure that the record companies still obtain revenues, it is important that the developers in the post-Napster era create commercial alternatives to the user-driven free beer networks. With the right commercial package, I am certain that record companies and artists can find a future in the post-Napster era without monitoring everything in the private sphere. After all, the fact that the record companies would stay away from my hard drive wouldn't mean that they waive all rights to digital music.

7 The future of intellectual property

Communication is important, and no matter what your favourite lobbyist and favourite lawyer tell you, technical progress and innovation should not be sacrificed on the altar of copyright. We need a balance between users and authors where Tim Robbins' character in Antitrust has good incentives to innovate, but where society at large is not too restricted due to Robbins' previous innovations. We also need a copyright commons where innovators may innovate and create without having to call their lawyer before they strike a chord on the guitar.

All this may sound easy to agree upon in theory, but in practice these propositions raise a lot of important questions. What should you do with current intellectual property proprietors? How will you keep incentives for very costly types of innovations, like drugs, computer programs and big screen movies? In theory, it is easy to stifle innovation through limiting copyright protection, regardless of area. In practice, it is more complicated as the case for "lagom" copyright illustrates.

The conversation continues.