Journal of Information, Law and Technology

Making Copyright Ambidextrous: An Expose of Copyleft

Maureen O'Sullivan
Teaching Assistant
University of Warwick

This Article is based on a revised and updated chapter from an LL.M Thesis, submitted at the University of Warwick, September 2001.

This is a **refereed** article published on: 6 December 2002

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

Abstract

The phenomenon of free or open source software (OSS) has garnered increasing attention in the legal field over the past number of years. It provides a paradigmatically different model of software development and marketing than proprietary software, which has traditionally been protected by copyright, and latterly, also by patent law. Licensing styles of free or OSS vary greatly from the very permissive, where users can privatise their modifications, to the quite restrictive, where programmers are obliged to contribute any changes they make to a communal software pool, which forms a species of expanding virtual commons. Examples of the former include BSD Unix licences and of the latter the GNU General Public Licence (GNU GPL), well known for being the licence used for the Linux operating system. This Article distinguishes between free software and OSS, discusses free and OSS licensing, comparing a BSD licence with the GNU GPL in order to illustrate the varying parameters which different programmers put in place to protect their programs. It also analyses the efficacy of the GNU GPL both from strictly legal and broader socio-legal perspectives. It concludes that this licence has facilitated an efficacious and productive management of what could otherwise have turned into an obsolete and deficient commons.

Keywords: Free software; open source software (OSS); copyleft; GNU General Public Licence; Linux.

1. Introduction

Free or open source software (OSS) has become widely used in the last few years, along with the expansion of the Internet. This Article gives a brief history of the development of free or OSS, seeking to clarify their differences, which are primarily of a philosophical nature. It then describes the attractions of this software and discusses different licensing regimes currently in use, focusing on the most popular licence of all: the GNU General Public Licence (GNU GPL). It analyses legal and socio-legal issues relating to the GNU GPL and charts the increase in its use among free or OSS programmers.

2. History of Free Software and Open Source Software

When operating systems such as Unix were first developed, it was normal practice for programmers to share the source code with few restrictions. The American telephone company, AT&T developed Unix and originally gave it away almost for free because it was only allowed to profit from the telephone network and not from any software which it developed. Computer researchers, particularly at the University of California at Berkeley, had added features to Unix and fixed bugs since its inception, and their version of Unix became known as BSD Unix, which was freely distributed for non-commercial purposes. When AT&T's monopoly was broken up in the early 1980s, it was able to profit from its version of Unix. As its price rose, its source code, or the 'crown jewels' of the operating system, was no longer available to be modified and instead was kept secret by the company. The trend towards privatising software became popular during this period.

Richard Stallman is an eminent hacker, who claims that:

> '[s]haring of software … is as old as computers, just as sharing of recipes is as old as cooking … [w]e did not call our software 'free software', because that term did not yet exist, but that is what it was'.

He resigned from his job at Massachusetts Institute of Technology Artificial Intelligence Laboratory (MIT) when this institute privatised its software and he started developing a non-proprietary, Unix-compatible operating system, known as 'GNU', a recursive acronym for 'GNU's Not Unix!'.

Subsequently, he founded the Free Software Foundation (FSF), a software charity whose mission is the development of software with accessible source code, which Stallman terms 'free software'. He believes that programming freedom includes being able to run the program for any purpose; having the liberty to modify the program, which necessitates access to the source code; being permitted to redistribute copies either for a fee or gratis; and being able to distribute modified versions of the program so that other hackers can benefit from the additions.

Some time after Stallman started developing GNU, a hacker called Linus Torvalds was taking a course in Unix at Helsinki University and he developed a kernel, which he named Linux. GNU's kernel, 'Hurd', was incomplete and Linux lacked programs so the latter was tweaked to fit the former and the Xfree Consortium graphical windowing system was added. Linux is a new operating system, which used no Unix code in its creation. It was built incrementally and the official Linux 1.0 was ripe for release in March 1994. Some US academics argue that the product should be known as GNU/Linux and at one stage there was an attempt by the FSF to rename 'Linux' as 'Li*gnu*x'.

The attention paid by hackers to semantics has had a significant effect on what was first denominated the 'free software' movement. In 1998, some hackers decided to coin a new phrase for 'free software' and renamed it 'open source software', believing that this would be more receptive toward commercial firms, which had begun to show an interest in adding to this software. Not everyone in the free software camp followed suit, however. Stallman describes the focus of open source as being about how to make high quality software but as far as he is concerned, the freedom to use and modify software is more important. This is neatly expressed in the following quote: 'Free Software is a political philosophy; Open Source is a development methodology'. For Stallman, the battle begins and ends with the fight to keep source code free: itself a lofty aim. NASA, an enthusiastic Linux user has declared 'software is not software without its source code'.

According to a leading voice in the open source software movement, Eric Raymond, open source advocates are also concerned about rights and freedom but only when they judge it to be appropriate. They do not reject Stallman's ideals but believe that his rhetoric and tactics will lead to failure in the commercial world. The term 'free software', while always painstakingly defined by Stallman as 'free as in speech, not as in beer', has been deemed both imprecise and antagonistic towards the corporate world. 'Open source' thus seeks to avoid ambiguity. The war of words rages on and an initiator of the coinage of the new term, Bruce Perens, now believes that the FSF's efforts have been eclipsed, despite his best efforts.

Regardless of whether one refers to this software as 'free' or 'open source', it is becoming increasingly ubiquitous and is challenging the hegemony of proprietary software in some fields, while it dominates in others. Its mode of development is ideally suited to the infrastructure of the Internet and, indeed, Linux, as with other OSS, is created largely by a loose, international community of hackers. The expansion of the Net has facilitated the growth of other OSS such as Apache, which has captured 61% of the Web server market; the Practical Evaluation and Reporting Language (Perl), used to write Web based applications and known as the 'duct tape' of the Internet; and a popular email server product called Sendmail, which routes over 80% of all Internet email messages. The domain name system is almost entirely dependent on the OSS program, Berkeley Internet Name Daemon (BIND), which allows users to seek textual rather than numeric addresses, for example: <http://www.altavista.com>. FSF core development tools such as compilers and debuggers pervade the software industry and the World Wide Web software was distributed free of restrictions. Yahoo!, the Net's largest website, is built around a free Unix version, FreeBSD, Apache, and Perl. In a world without OSS, the Internet would virtually grind to a halt. Lessig endeavours to capture this software's ubiquity when quoting an engineer acquaintance of his who exclaimed:

> '[T]he 'ah-ha' for Open Source Software came to me when I realized, 'wait, open source is the Internet''.

The manner in which free or open source software is developed greatly simplifies onerous tasks such as debugging and this speeds up the pace of releases. It also leads to more robust software because it is meticulously reviewed. In fact, Linux has achieved an almost mythical reputation for never crashing, distinguishing it from less reliable proprietary software, which does not have the benefits of such widespread peer review. Raymond believes than when an essential product is being tested, 'the more testers the merrier' should be the motto. In order for such review to take place, source code is best kept open.

3. Licensing Largesse

Hackers generally choose not to place unlicensed software in the public domain, with a few exceptions, such as the World Wide Web (WWW), whose inventor, Tim Berners-Lee's principal concern was to disseminate his invention and he believed that copyright restrictions would interfere with this aim. Most hackers, however, wish to retain some non-exclusionary form of intellectual property right over their work. What normally distinguishes commercial and OSS licences is 'the presence or absence of certain licence terms' rather than the presence or absence of a licence.

Free Unix versions have never been placed in the public domain but are released to the public under very generous licensing terms. BSD-style licenses are considered to be the most liberal, in the sense that BSD code can be included in proprietary programs. This licensing style includes licences, all with similar conditions, such as X-type licenses, BSD and modified BSD (also known as old and new BSD licences) and the Apache software licence, to name but a few.

In contrast, the GNU GPL, which is used to license the Linux operating system, is a species of licensing denominated ‘copyleft’, often found both in the free and OSS camps, and this term includes some other licences of lesser repute. Unlike it mirror image copyright, which reserves all rights, copyleft jokingly promises to reverse all rights. The principal copyleft licence, the GNU GPL, was devised by Stallman.
Torvalds did not originally choose the GNU GPL to license Linux but after a short period decided it was the most suitable as the one he had originally selected allowed the user very little freedom.

The Open Source Definition is a set of guidelines for OSS licences drawn up by the Open Source Initiative, a non-profit corporation. This corporation provides an OSI Certified Open Source Software certification mark, which can be used if the chosen licence is OSD compliant. Both BSD licences and the GNU GPL conform to the OSD but of all the licences covered by this definition, these are the two most diametrically opposed due to the differing philosophies and commercial foci of their proponents. According to some surveys, they are also the two most commonly used licensing styles.

3.1. Laissez-Faire Licensing, Bsd-Style

BSD licences provide for the release of university-developed software, the source code of which is not kept secret. The researcher and institution must be acknowledged and copyright notices maintained, which is a species of moral right to the software. Permission must be obtained should anyone wish to use the author’s name to endorse derivative works, which protects against false attribution. A rather unpopular provision, which still survives in old BSD licences but which has been removed in newer ones, was the onerous requirement that the authorship of the software be acknowledged in advertising material. Use of older licences is now discouraged. The code may be modified but the original developers do not usually accept contributions from the public and forking of the code base is permitted. Thus, BSD Unix can be used to create proprietary derivative works of which the source code is not made available. This provision has promoted the proliferation of BSD Unix in the building of the Internet, and this style of licence governs other open source products such as BIND, Apache and Sendmail. As is common practice, the licence also contains a disclaimer of warranty, which does not practically distinguish it from commercial software. A typical new BSD licence is now reproduced.

> Copyright © 1998 <SOMEONE>
> All rights reserved.
>
> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
>
> 1. Redistributions of source code must retain the above copyright notice, this list

> of conditions and the following disclaimer.
>
> 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
>
> 3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.
>
> THIS SOFTWARE IS PROVIDED BY THE AUTHOR ‘AS IS’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BSD Unix, like Linux, is an open-access commons: with a difference. Whereas software is not prone to depletion from overuse, if code used in derivative works is never contributed back to the community, the software commons may be prone to obsolescence and could thereby be rendered useless, if its development team discontinue its maintenance. In an interview with *Infoworld*, Bob Young, a former CEO of Linux’s biggest vendor, the company Red Hat, was asked whether Linux companies, which bundled proprietary software with the free operating system were likely to succeed in the market, to which he made the following reply:

> ‘Our perception is that the reason that Slackware and Debian are the second and third most popular versions of Linux behind Red Hat is because they’re also delivering the same control benefit to users. The problem with other, smaller commercial Linux vendors is they look at Linux and they see…[g]reat technology, but a broken economic model. So…they take Linux and surround it with proprietary tools…[F]rom a support and bug tracking issue, you’ve effectively just bought another proprietary binary-only OS’.

Despite BSD Unix’s repute, Microsoft has acknowledged that Linux is impinging on its market and will eventually dominate it entirely.

Brian Behlendorf, one of the founding members of Apache, uses a BSD-style licence but acknowledges that companies who use their software need to reinvest in order to help Apache grow. However, this is done on a voluntary basis and does not compel any contribution from any firm, which uses the code. On the other hand, the GNU GPL seeks a more specific contribution and does not make this optional.

3.2. Gnu Gpl: Creating A Contributory Commons

In order to avoid sharing the fruits of his toil with those unwilling to do likewise, Torvalds originally released Linux under a licence which permitted redistribution, on condition that that the source code was provided free of cost. Copyright notices had to be preserved, although there was a fair use provision, which allowed for the reproduction of ‘small partial excerpts’ without any copyright acknowledgement. However, Torvalds prohibited the charging of any sum, even if only intended to cover ‘handling’ costs, such as the inclusion of copies with the kernel. This term was unduly restrictive and could have hindered the breadth of Linux’s user base.

Torvalds finally chose the GNU GPL because it allowed redistribution while obliging users to contribute any modifications they had made back to the software commons. This facilitates the creation of a special type of commons, unparalleled in the physical world, referred to by Raymond as the ‘inverse commons’ and by Rose as the ‘comedy of the commons’: that is, a commons, which grows and expands with use rather than shrinking or wearing out.

The GNU GPL allows modification and sale of the product but such derivative works must be donated to the open source software pool. This helps to avoid forking and has facilitated the growth of firms, which sell open source software. Furthermore, the source code of any such works must be provided for free. If GPL-ed code is used together with a proprietary code, the source code of the latter must also be revealed. The licence has thus been described as ‘a virus subverting the proprietary ‘host’ program to create more GPL-ed code’. It has been said that:

> ‘There are users who say that … the GPL ‘excludes’ some proprietary software developers who ‘need to be brought into the free software community’ … . Their decision to make software proprietary is a decision to stay out of our community. Being in our community means joining in cooperation with us; we cannot ‘bring them into our community’ if they don’t want to join. What we can do is offer them an inducement to join. The GNU GPL is designed to make an inducement from our existing software: ‘If you will make your software free, you can use this code’.

Of note is the aforementioned fact that Linux is proving to be more successful than BSD Unix which has endeavoured to maximise the freedom of commercial firms not obliged to release the source of BSD Unix-derived works. Lessig describes the GNU GPL as ‘Stallman’s real genius’ and says that it is a licence, which uses ‘the power of copyright to guarantee that what was produced under GPL not be removed from the commons’. The licence is reproduced in Appendix A.

3.3. Quasi-Constitutional Facets of the GNU GPL

The GNU GPL is a rather loquacious licence, which includes an aspirational preamble closely resembling that of many constitutions and laden with moral prescription. Whereas constitutions often profess to take their cue from a heavenly body, the GNU GPL hones in on a devilish icon to be eschewed at all costs: that being the archetypal proprietary software licence. This licence presents potential signatories with an all-or-nothing proposition: acceptance of the terms has a knock-on effect for any future generation of assignees.

Open source society’s consciousness, according to Raymond, did not begin to stir until the early 1990s and Stallman played an important role in raising its cultural awareness. One of the reasons for the writing of this ‘constitution’ arose from the demonisation of hackers, which served as a catalyst for a type of declaration of independence from the proprietary software world and all that it engendered. The Preamble holds as its ideal the freedom of its users and ‘territory’ from colonisation. It is an assertion of the sovereignty of open source participants against those who refuse to reveal their source code and appears to operate within a defined on-line territory.

Raymond expresses his concept of the community in language analogous to that of a state, albeit a virtual one. Open source territory is not cyberspace: on the contrary, it exists in ‘the space of all programs’. He quotes the words of a hacker, Fare Rideau, who says that hackers own:

> ‘programming projects which carry with them inter-social bonds’.

Programming space is akin to a software commons and projects constitute the ‘property rights’ in this arena. It is an unusual form of territory, however, in that entry is unrestricted and one may partake of any of its fruits, at no cost. The only prohibited acts, the forbidden apples as it were of this realm, are to exclude others from accessing and modifying one’s own contribution, to feign authorship of another’s program or, indeed, to attribute wrongfully to another the authorship of one’s own work - Biblical-style prohibitions whose violation could well result in e-exile. In addition to its quasi-constitutional status, copyleft also purports to rely on copyright law to enforce its strictures.

3.4. The GNU GPL’s Multidimensional Legal Protection

At the beginning of the licence, readers are advised that while copying and distribution of the text is not prohibited, the licence, unlike its subject matter, may not be modified.

The first three paragraphs of the Preamble of the GNU GPL explain the purpose of the licence and indicate that some responsibilities are expected in return for privileges granted.

Paragraph 4 of the Preamble informs users that the distribution of copies of the program entails an obligation to convey all rights received to transferees, including information regarding the licensing terms, a condition which is reiterated constantly throughout the body of the text.

The fifth paragraph pledges the dual protection of copyright and contract, and copying, distribution and modification of the software are permitted. The first clause of this paragraph relating to copyright is rather ill-phrased, in that it does not make clear in whom the copyright is supposed to vest. What may be intended is to provide an assurance that the software owner genuinely possesses the copyright in order to allay concerns that its use and modification will infringe someone else’s rights. Much of the copyright in derivative works belongs to the FSF, having been assigned by the authors of modifications. However, if such an assignment is not performed, the author of the derivative work may seek to claim an independent copyright but the GNU GPL was designed to protect against such action. If someone were to copyright a derivative work under the misapprehension that the software lay in the public domain, perhaps due to a failure to receive a copy of the licence, the GPL would not be violated but there would be a copyright infringement. This is due to the fact that a lack of awareness of extant copyright is no defence to a breach: copyright therefore purports to operate as a backup of the licence. McGowan points out that a fair use defence may be mounted to a charge of infringement if users copy code in order to make the program compatible with their own. The GNU GPL presents the reader with a certain ambiguity in this regard: in the Preamble, paragraph 4 it pledges to allow users to pass on all their rights and in paragraph 5 states that rights are protected both under copyright law and the licence itself. It does not state whether use of the code encompasses all the fair use rights a user would have under copyright law or whether copyleft endeavours to abrogate exceptions to the strictures of copyright.

The next paragraph of the Preamble, like BSD licences, disclaims all warranties for the software, which is justified on the basis that its cost is minimal. This may fall foul of the law in certain European countries, however. 1999 saw the founding, in Germany, of the Institute for Legal Questions of Open Source Software, which is a forum where lawyers can discuss issues such as the GNU GPL's enforceability in German law. It was concluded that this term would not be upheld, especially if the software was for sale. Another term in this section is that modified software must be identified as such, in order to guard the original developer's reputation. This is a protection against false attribution which, given the significance of reputation in the open source community, is of primary importance.

Paragraph 7 highlights the threat that patents pose for free software and advises that any re-distributors of the software who obtain patents must license it for everyone's unhindered use or refrain from licensing it at all.

Following the Preamble, the terms of the licence are listed. Section 0 states that the licence may be applied to works other than software and it is especially suitable for creations of an academic nature from which the author is unlikely to benefit economically but may nonetheless wish to earn reputation.

The first section permits literal copying but copyright notices and warranty disclaimers must be maintained, although additional charges can be made for the provision of a warranty. Licensees are entitled to charge when transferring a copy of the software and there is a reiteration of the requirement to distribute the licence when doing so.

Section 2 permits the modification and distribution of the software but new licensees must be informed, by way of notices, of any changes made. Derivative works must be licensed in their entirety and no charge may be made for these works. Section 2, paragraph 2 has provoked some controversy because if a GPL-ed program contains parts which are not based on OSS but which are included in a derivative work, copyright claims may not be made in relation to these independently created works. This proviso is to ensure the maintenance of the copyleft inclusionary boundaries but obligates the surrender of proprietary rights to any additional code. A difficulty with the implementation of this section is that even under US copyright law, what constitutes a derivative work may be the subject of uncertainty. Furthermore, there appear to be inconsistencies between what the FSF classify as a derivative work and how such a work is defined under copyright. Haynes criticises this provision, claiming that:

> 'even the open source movement exploits and is hindered by copyright… users of open source risk losing their own copyrights … [t]hey lose the benefits of the intellectual property system designed to protect their investment'. However, he misses the point that investment costs are greatly reduced by being able to use free code and that by becoming part of the free or open source community, bug-tracking and peer review are both available free of cost.

Users are not prohibited, under section 3, from distributing the program in object code, provided that the source code is made available or an undertaking is given to provide the source code, presumably on request. If the re-distributor has only received the program in object code, information as to where to obtain the source code must be imparted.

The fourth section provides that third party rights are not invalidated if the original licensee breaks the terms of the licence, providing that these are adhered to. If the contractual aspect of copyleft is not upheld, any claims for redress hinge on copyright, which vests automatically in the original creator. This shows a subversive use of copyright, which allows the general public to benefit from a relaxation of its stringent rules, while ensuring that no one takes advantage of this freedom to exclude others. However, the present penalty for copyright infringement would not necessarily entail an obligation to reveal the source code of any modifications to the wronged party nor to hand over possession of any derivative works, so copyright law as it stands is not an ideal solution to deal with violators of the licence. Moreover, on a global scale, it is possible that an individual or organisation's copyright will not be recognised uniformly throughout the world. It is likely, however, that any such code along with its developer, would be rejected by the community and any such modification would have to be maintained in isolation, thereby losing the benefits of peer review and bug-tracking. This would only directly affect violators who are members of an OSS group but negative publicity associated with non-conformity to OSS community norms may deter offences of this nature.

Section 5 informs licensees that by accepting the terms of the licence, they are bound by it. A potentially controversial issue which arises is the manner of distribution, especially if the licence is in shrink-wrap or click-wrap form. Such licences have been upheld in the US in *ProCD, Inc. v. Zeidenberg* and *Hotmail Corp. v. Van$ Money Pie Inc* respectively but licensees must be aware of the existence of the contract for it to bind them. These licences have become enshrined in legislation in the US by virtue of the Uniform Computer Information Transaction Act (UCITA), 1999 (Amended 2000, 2001), with some provisos. Assent to any terms must occur after the user has had a chance to review them and payment must follow any such review. The right to reject the contract must be cost free but if the user does so, he or she rescinds the right to use the software. To date a version of this Act has been passed in a number of states and will eventually be considered by all.

The next section of the GNU GPL asserts that redistribution entails the receipt of a licence from the original licensor to copy or modify the program and that further restrictions cannot be imposed. The user is not responsible for enforcing the compliance of third parties under this section so a violation by a licensee does not make the licensor liable for any damages.

Should any part of the GNU GPL be struck down, Section 7 of the licence provides that the balance of the section should apply, thereby preserving its overall spirit.

The eighth section stipulates that a geographical region may be excluded from receiving the software if its use there is inhibited by patents or copyrights.

Under Section 9, the FSF reserves the right to alter the licence in order to address new circumstances which may arise. The advantage of this provision is that the licence can be adapted quickly to evolving technology and, in fact, the current GNU GPL is version 2 of the original licence. However, individuals are presently free to use any version of the GNU GPL they choose.

Despite the fact that the GNU GPL is over a decade old, it has yet to make its court debut due to an apparent universal compliance, once reminded of its terms, by all who have attempted to breach the licence. This suggests that whereas its legal validity is untested in the US and has been deemed to be of partial applicability only in some EU states, such as Germany, its socio-legal status is well established and merits examination.

3.5 GNU GPL: A Socio-Legal Analysis

Professor Moglen, who is the FSF's general legal counsel, says that approximately twelve times a year, a GNU GPL violation of FSF copyrighted software comes to his attention. To date, no recourse to the courts has been made: instead his method of dealing with these infringements is to contact the licensees, inform them of their contravention and compliance is invariably attained. This is a clear example of practices in close-knit communities whereby reports of any rule infractions are made by watchful neighbours, in this case hackers, and dealt with swiftly and informally.

Moglen comments:

> '[N]o large American software company has engaged in a public controversy with us over the enforceability of the GPL … some might conclude 'that means…there's something about the GPL [that] is not enforceable', I would turn that proposition around … there have been no such controversies because nobody thinks they're going to win them'.

There are some echoes in this description of *Lex Mercatoria* or the Law Merchant in that its rules were based on ‘good faith’; the law did not play a part in ensuring compliance; and

> ‘[r]eciprocity and the threat of business sanctions compelled performance’.

The equivalence of business sanctions in the open source world involves the threat of damage to a firm’s reputation and co-operative behaviour is achieved, if not voluntarily, through the shunning and flaming of offenders. The law may, however, have created expectations regarding likely outcomes of disputes where custom has not been observed. The GNU GPL adds an extra dimension to such chastisement: the possibility of a lengthy and costly court case, together with negative publicity, which would likely ensue.

Intimation of imminent legal action, which often does not materialise may be a prelude to negotiation and settlement rather than a court hearing. This practice has been described in the family law arena as ‘bargaining in the shadow of the law’. In such scenarios, by the time the case reaches the courts, frequently the only matter remaining to be resolved is formalisation of the agreement by means of a court order. Similarly, when Moglen contacts offenders, the GNU GPL reinforces his words, should any appeals to respect hacker customs be disregarded. This suggests that the processes at play in ensuring compliance with the terms of the GNU GPL exhibit features of legal pluralism, which, to borrow from open source terminology, is law in a ‘forked’ state. If one remedy is not effective, then another may be and there are several *quasi*-legal forks from which one can elect. Palmer and Roberts describe this selection as an evolving phenomenon of a ‘pick-and-mix’ approach to dispute resolution, aimed at performing this task with greater skill.

Clark and Economides argue:

> ‘we have to recognize that the practice of law is a social activity encompassing a multitude of tasks which are only partially guided by formal legal rules … and that many of these ‘legal’ tasks can be and increasingly are executed by individuals working in organizations which are not subject to the guardianship and control of the legal profession’.

Nonetheless, Moglen believes there is an urgency to test the cognisability of the GNU GPL in the courts, presumably to enshrine informal hacker customs in the common law.

3.6 If It Ain’t Broke, Why Fix It?

It is not entirely clear why Moglen, who has almost single-handedly cowed large corporations into compliance without even so much as a writ, deems litigation to be desirable, given that his present method of enforcement has proved to be 100% effective to date. If enforced by the courts decisively, however, this would undoubtedly bring great relief to those interested in maintaining the vivacity of free and OSS and reduce some of the time involved in informal policing. Raymond, although claiming the following view to be a minority position within the hacker community, says he believes that:

> ‘the GPL is much more important as an implicit social contract than as a legal document’.

However, he considers that:

> ‘most hackers don’t trust the superiority of open-source development to want to give up using force of law to make it happen’.

Moody speculates that if the GNU GPL is not upheld in court, the open source world would be severely affected, so perhaps it would be better for the community if prospective court appearances were held in a state of suspended animation. Furthermore, litigating the GNU GPL, even if successful, may well undermine hacker confidence in their own customs and convert their community from a *gemeinschaft* into a *gesellschaft* society by relying on extraneous, impersonal legal processes over which they have no control.

The idea of the superiority of law as a means of regulating conduct appears to be firmly entrenched in open source quarters. However, in the analogous proprietary software world, legal measures against ‘piracy’, which is the nearest thing to a GNU GPL violation in the open source sphere, have proved hopelessly inept at containing this practice. A 1998 Price Waterhouse Report, commissioned by the Business Software Alliance estimated that over 40% of European business software was illegally copied. Curiously, they deem the alleged US average of 27% to be a manageable rate and thus have no expectation of attaining a level of compliance comparable to that achieved by the as yet judicially untested GNU GPL, despite intensive lobbying for increasingly punitive laws by which they hope to protect their intellectual property.

Law is often described ‘in terms of legislature, rules, courts and enforcement agencies’, the legislature is perceived to be the ‘ultimately authoritative’ source of law, and its rules uniquely ascertain the result of disputes. An unfortunate outcome of this type of system is that compromise is not encouraged, as it requires judges to determine rather than mediate issues before them. Parnell describes a judicial district of southern Mexico, in which inhabitants can choose between social rules and court processes which both co-exist for the resolution of certain disputes. A counsellor in this region states that:

> ‘once a dispute reaches the district court … there is no ‘forgive me’’,

indicating that the court process has the effect of polarising the respective positions and that compromise is more likely to be achieved by means of social rules.

In any case, if the courts are not favourable to the ideology of the GNU GPL, it may not be upheld. Lessig says that:

> ‘[t]he law prefers opaque to transparent code; it constructs incentives to hide code rather than to make its functionality obvious … [o]ur law creates an incentive to enclose as much of an intellectual commons as possible. It works against publicity and transparency and helps to produce, in effect, a massive secret government’.

Given such a climate, litigating the GNU GPL may prove to be too much of a gamble and the enforcement of this type of law is more likely to be assured if first enshrined in statute, which is not entirely improbable, given that its ethos has come to the attention of a number of prominent bodies around the world.

3.7 GNU GPL: Acquiring a Lex Mercatoria-Style Force of its Own?

The GNU GPL's doctrine is gaining ground, certainly in OSS camps. The traditional software paradigm of software as a product is becoming less popular and its value is increasingly to be found in the services that are provided as adjuncts. Copyleft provides a better software distributive model than the proprietary one in that it ensures that a source code commons remains available. Philippe Queau, who is the director of the Information and Informatics Division of UNESCO says that the organisation is aiming to promote the public domain over privatisation, that copyleft can help this endeavour in that it encourages authors' recognition rather than monetary reward, and that UNESCO is proposing that copyleft be made a positive right. Copyleft does not, in fact, rule out monetary reward but, rather, indulges authorial reputation by permitting more participation in collaborative projects, while at the same time allowing the sale, but not the enclosure, of the software.

In 2000, there was a proposal in France of a law, which would have made the use of software:

> 'of which the use and modification are free and for which the source code is available',

mandatory throughout French public organisations. An Agency of Free Software was set up in order to study appropriate licensing schemes. However, the idea was not ultimately adopted as it was perceived to pose a threat to competition. Subsequent, more limited proposals suggest making source code available solely to the government.

Argentina has considered a number of bills, which would make it mandatory for all government offices to use open source software. Stringent penalties for piracy of computer programs were introduced in 1998, after which a 45 day period of grace was granted during which time violators had an opportunity to emend their status. At the expiry of this time, Software Legal, the Argentine equivalent of the BSA, zealously targeted persistent offenders at which juncture less than half chose to comply. Among repeat reprobates looming rather large on the piracy blacklist are government departments such as tourism and social security who are tiring somewhat of being tarred with the piracy brush. Their consideration of open source has not gladdened the hearts of Software Legal, however, which now has cause to fret over the fact that their chances of competing may be curtailed. Other Latin American countries such as Peru are also seriously considering promulgating the use of OSS in government administration but are being challenged by representatives from Microsoft who are not especially thrilled at this prospect.

The German Bundestag has deliberated OSS, arriving at a unanimous vote:

> ‘OSS should be used and promoted throughout the administration’.

Of particular concern is the fact that despite present EU limitations on the granting of software patents, any future relaxation of this stance might result in a chilling effect on any further development of free software. Another focus was on the need to keep encryption methods open to ensure security, as when source code is not accessible, the existence of ‘back doors’ or deliberately collocated software holes which facilitates monitoring, is impossible to ascertain.

A more significant development has occurred in the European Community where the Information Society Directorate General of the European Commission has set up a Working Group on what is referred to as ‘libre software’, meaning both free and open source software. The remit of this group is to study the libre software phenomenon and to draw up a set of proposals for the Community, which has been published as a work in progress. The group is critical of liberal US policy on the granting of patents, which it opines will hinder the development of proprietary as well as OSS, and may lead to a plethora of inadvertent intellectual property infringements. As such, it advocates that software patenting should be resisted to ensure that it does not become a weapon against OSS development. The group perceived this software as:

> ‘both a great opportunity and an important resource’ ,

and advised that the recommendations should not be understood as helping open source software, but rather how best:

> ‘to help Europe to benefit from open source software’.

The IDA (Interchange of Data between Administrations), a body set up to facilitate OSS use throughout the EU recently commissioned a report entitled ‘Pooling Open Source Software’, although such software may be released under a different licence, and not the GNU GPL. This is an Esperanto-type endeavour to develop a climate of sharing bespoke software, which is used more often in the EU than OSS, and when not shared, is much more costly.

A factor likely to follow hot on the heels of any such measure is new legislation, of which the spirit of the GNU GPL should be at the core. This kaleidoscopic socio-legal subterfuge may well be incorporated into European law and elsewhere in some form in the future. If governments begin to adopt OSS, after some time they will wish to ensure that they too profit from any modifications they make. Of prime importance will be the close tailoring of any such law to hackers’ customary rules. If this is not carried out effectively, there lurks the possibility that the open source commons will be destroyed.

The growing prominence of OSS is inextricably entwined with the Internet’s expansion. In this new medium, protecting software as a product poses an inordinate juxtaposition with the type of coverage actually necessary. The pace of Linux’s market penetration is likely to continue and, far from being ‘hackerware’, this operating system has gained mainstream respect.

Hackers gleefully quote Gandhi when postulating on their future fate.

> ‘First they ignore you. Then they laugh at you. Then they fight you. Then you win’.

4. Conclusion

Open source software may be licensed in a number of different ways. Users may be permitted to do whatever they want with the code, a few minor restrictions aside, or they may be able to participate in a less selfish way by being obliged to contribute back to the software commons the source code of any modifications they have made. Without some onus on users to give something back for what they get, it is unlikely that a robust software commons will continue to flourish indefinitely. The GNU GPL, which is the licence most often used in this realm, has been central to OSS’s success. Any legislation drafted and implemented by governments considering the use of this software should be based on this licence. Governments should be guided by what has worked for a dozen years in the open source realm and not attempt to place too many or too few restrictions on this resource unless they are prepared to play dice with its survival. The have chosen to move away to an extent from proprietary models of software and should be open to consider new or unfamiliar forms of intellectual property protection which have a proven track record in open source terrain.

Appendix A

GNU GENERAL PUBLIC LICENCE
Version 2, June 1991
Copyright © 1989, 1991 Free Software Foundation,
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies
of this licence document, but changing it is not allowed.

Preamble

The licences for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licence is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public Licence applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public Licence instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licences are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. Any you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this licence which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

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

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENCE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This Licence applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public Licence. The 'Program', below, refers to any such program or work, and a 'work based on the Program' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term 'modification'.) Each licensee is addressed as 'you'.

Activities other than copying, distribution and modification are not covered by this Licence; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this Licence and to the absence of any warranty; and give any other recipients of the Program a copy of this Licence along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

(a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

(b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this Licence.

(c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this Licence. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this Licence, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this Licence, whose permissions for others licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this Licence.

3. You may copy and distribute the Program (or a work based on it, under Section 2)

in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

(a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for non-commercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection (b) above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this Licence. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this Licence. However, parties who have received copies, or rights, from you under this Licence will not have their licences terminated so long as such parties remain in full compliance.

5. You are not required to accept this Licence, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this Licence. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this Licence to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the

recipient automatically receives a licence 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 granted herein. You are not responsible for enforcing compliance by third parties to this Licence.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this Licence, they do not excuse you from the conditions of this Licence. If you cannot distribute so as to satisfy simultaneously your obligations under this Licence and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this Licence would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public licence practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licence cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this Licence.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or copyrighted interfaces, the original copyright holder who places the Program under this Licence may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this Licence incorporates the limitation as if written in the body of this Licence.

9. The Free Software Foundation may publish revised and/or new versions of the General Public Licence from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this Licence which applies to it and 'any later version', you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this Licence, you may choose any version ever published by the Free Software

Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the 'copyright' line and a pointer to where the full notice if found.

> <one line to give the program's name and a brief idea of what it does.>
> Copyright © 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A

PARTICULAR PURPOSE. See the GNU General Public Licence for more details.

You should have received a copy of the GNU General Public Licence along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.
If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright © 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical command ‘show w’ and ‘show c’ should show the appropriate parts of the General Public Licence. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items-whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a ‘copyright disclaimer’ for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public Licence does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public Licence instead of this Licence.