THE JOHN MARSHALL
REVIEW OF INTELLECTUAL PROPERTY LAW

WHY PAY FOR WHAT'S FREE?: MINIMIZING THE PATENT THREAT TO FREE AND OPEN SOURCE SOFTWARE

KIRK D. ROWE

ABSTRACT

The growing popularity of free and open source software as a viable alternative to proprietary software has made it an unwitting participant in an inevitable intellectual property law confrontation that will pit patent against copyright. Where proprietary software is primarily protected by patents, which seek to exclude others from the use of specific ideas, open source software utilizes a variation of copyright protection, which seeks to force the inclusion rather than the exclusion of third parties' access to expression. Because these methods of protection are as different as the software models themselves, it is difficult to predict the outcome of this unfolding conflict which has never been directly litigated in court. This comment advocates that the public would be best served by allowing both software models to coexist. From this position, it explores the distinction between the two types of protection utilized and recommends changes which will both maximize innovation and preserve freedom of choice.

Cite as Kirk D. Rowe, *Why Pay for What's Free?: Minimizing the Patent Threat to Free and Open Source Software*, 7 J. MARSHALL REV. INTELL. PROP. L. 595 (2008).

WHY PAY FOR WHAT'S FREE?: MINIMIZING THE PATENT THREAT TO FREE AND OPEN SOURCE SOFTWARE

KIRK D. ROWE*

INTRODUCTION

In 1991, when the Microsoft Corporation had fewer than fifty filed patent applications, CEO Bill Gates stated in a memorandum:

> If people had understood how patents would be granted when most of today's ideas were invented, and had taken out patents, the industry would be at a complete standstill today . . . . I feel certain that some large company will patent some obvious thing related to interface, object orientation, algorithm, application extension or other crucial technique.[1]

Since then, Microsoft has filed an average of over two patent applications per day, and similar numbers hold for other large software companies.[2] Can this much innovation really be taking place, or is there something else at play here?

In this modern age of computing, free alternatives to proprietary software exist for just about every need.[3] Because free and open source code is widely accessible for software developers to examine and improve upon, its rapid evolution is unparalleled by most conventional commercial software.[4] Corporate software giants fear that the market for their software will evaporate.[5] For instance, consider Linux,[6]

* J.D. Candidate, May 2009, The John Marshall Law School. The author would like to thank the editors of THE JOHN MARSHALL REVIEW OF INTELLECTUAL PROPERTY LAW, particularly Brad Nykiel, Zion Park, Tony DiVincenzo, and Tim Cho, for their hard work and welcomed suggestions.

** Available at www.jmripl.com.

[1] Memorandum from Bill Gates, Microsoft Board Chairman, Challenges and Strategy (May 16, 1991), *quoted in* Mark H. Webbink, *A New Paradigm for Intellectual Property Rights in Software*, 2005 DUKE L. & TECH. REV. 12, ¶ 3 (2005). Microsoft held only three software patents as of its fifteenth anniversary in 1990. *See* Timothy B. Lee, *Analysis: Microsoft's Software Patent Flip-Flop*, ARS TECHNICA, Mar. 13, 2004, http://arstechnica.com/news.ars/post/20070313-analysis-microsofts-software-patent-flip-flop.html (reporting results after conducting a comprehensive patent search for Microsoft patents).

[2] Webbink, *supra* note 1, ¶ 4. In the summer of 2004, Microsoft announced an increase in its annual patent filings from 2,000 to 3,000. *Id.* ¶ 6; Steve Lohr, *Pursuing Growth, Microsoft Steps Up Patent Chase*, N.Y. TIMES, July 30, 2004, at C3. As of 2005, Microsoft had over 4,000 patents with an additional 10,000 pending. Webbink, *supra* note 1, ¶ 4 (revealing results after searching Microsoft, as assignee, against the U.S. Patent and Trademark online database). IBM received between approximately 700 to 900 software patents annually between the years 2000 through 2004. William R. Haulbrook, *Getting a Handle on the Software Patent Explosion*, 7 PAT. STRATEGY & MGMT. 1, para. 14 (2005).

[3] Don Hardaway, *Replacing Proprietary Software on the Desktop*, 40 COMPUTER 3, 97–98 (2007).

[4] Greg R. Vetter, *Exit and Voice in Free and Open Source Software Licensing: Moderating the Rein Over Software Users*, 85 OR. L. REV. 183, 205 (2006).

[5] Henry W. Jones, III, *"Other" Software Licensing Pitfalls (Beyond Black-Letter Law Errors): Common Erroneous, Risky Assumptions of Clients & Counsel*, *in* UNDERSTANDING THE INTELLECTUAL PROP. LICENSE 2006, at 157, 176 (PLI Pats., Copyrights, Trademarks, & Literary

OpenOffice,[7] Firefox,[8] and Gimp,[9] which are open source software applications that can collectively replace proprietary operating systems, office suites, web browsers, and graphics editors. However, "big business" has made it clear it will not sit by idly as its profits wane.[10] Wielding patents, The SCO Group,[11] a Utah-based software provider, has already attacked the open source movement, and Microsoft is mounting its own infringement charge.[12]

This comment addresses the threat the recently recognized software patent poses to the free and open source software model and the associated defense strategy emerging to combat the threat.[13] Section I chronicles the emergence, practice, and philosophy of free and open source software. It reviews the history of open source software litigation and the changes that followed. Section II analyzes whether the expanded defenses to the patent threat are adequate, while section III proposes suggestions that may enhance those defenses. The concluding section summarizes some final thoughts.

Prop., Course Handbook Ser. No. 879, 2006), *available at* WL, 879 PLI/Pat 157. Apache, for example, has had the lion's share of the worldwide web server market for years. *Id.* MySQL, a database management system, and Samba, a networking protocol, are but two more examples of non-traditional market leaders. *Id.*

[6] Linux Online Home Page, http://www.linux.org/ (last visited May 20, 2008). Linux is a free Unix-like operating system that is the result of global collaboration. *Id.*

[7] OpenOffice: The Free and Open Productivity Suite, http://www.openoffice.org/ (last visited May 20, 2008). OpenOffice is an open source project by Sun Microsystems for a multiplatform and multilingual office suite that includes a word processor, spreadsheet, database, presentation software, drawing program, and equation editor. *Id.*

[8] Mozilla: Firefox 2, http://www.mozilla.com/en-US/firefox/ (last visited May 20, 2008). Firefox is a cross-platform open source web browser that represents the flagship product of the Mozilla project. *Id.*

[9] Gimp 2.4: The GNU Image Manipulation Program, http://www.gimp.org/ (last visited May 20, 2008). GIMP is an acronym for GNU Image Manipulation Program. *Id.* It is free software that allows for image creation and photo retouching. *Id.*

[10] Roger Parloff, *Microsoft Takes on the Free World*, FORTUNE, May 28, 2007, at 77, *available at* http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/index.htm. Microsoft wants royalties from distributors claiming that free software violates 235 of its patents but will not specify which patents. *Id.*

[11] The SCO Group, http://www.thescogroup.com (last visited May 20. 2008). The SCO Group was formerly known as Caldera Systems and created the now defunct Caldara OpenLinux, a business orientated distribution. *Id.* They are mainly a proprietor of the Unix operating system. *Id.*

[12] *See* Parloff, *supra* note 10, at 78.

[13] *See, e.g.*, Barbara Rose, *So Far, More Than a Match for Microsoft*, CHICAGO TRIB., Aug. 24, 2003, § 5, at 1 (providing an example of a software patent being leveraged against both proprietary and open source developers). In 1994, a team from the University of California, San Francisco, formed the company Eolas and filed a patent claiming the creation of the first web browser to support plugins. U.S. Patent No. 5,838,906 (filed Oct. 17, 1994) (issued Nov. 17, 1998); *see also* University of California Office of the President: Questions and Answers about UC/Eolas patent infringement suit against Microsoft, http://www.ucop.edu/news/archives/2003/aug11art1qanda.htm (last visited Oct. 26, 2007). Microsoft's use of this technology resulted in Eolas filing suit against them in 1999. Eolas Techs., Inc. v. Microsoft Corp., No. 99C0626, 2000 WL 1898853, at *1 (N.D. Ill. Dec. 29, 2000). Although not litigated in court, Eolas also seeks royalties from the open source community for alleged infringement. Eolas Licensing, http://www.eolas.com/licensing.html (last visited May 20, 2008).

I. Background

A. *Inception of Free Software*

Free software was the vision of hacker, developer, and activist Richard Stallman.[14] While working as a programmer for Massachusetts Institute of Technology's ("MIT") Artificial Intelligence laboratory in the early 1980's,[15] Stallman became frustrated when proprietary software began to replace the freely modifiable source code that was customarily provided with the computer hardware of the day.[16] Realizing that the software culture was changing to favor "big business," Stallman was determined to create an operating system that would remain open for all to control and modify.[17] Stallman abandoned his job at MIT in 1984 and set out to write a free Unix-like[18] operating system that would be identified with the recursive acronym "GNU" (short for GNU's Not Unix).[19]

In 1985, Stallman founded the non-profit Free Software Foundation ("FSF"), which advocates the freedom that Stallman believes software users are entitled to.[20] This free software movement, which is part of a larger open source software movement,[21] was built around the core concept of four freedoms of program use,

[14] Vetter, *supra* note 4, at 186–87.

[15] Interview by Richard Poynder with Richard Stallman, Founder, Free Software Movement (Mar. 21, 2006), http://ia310134.us.archive.org/1/items/The_Basement_Interviews/Richard_Stallman_Interview.pdf.

[16] Brian W. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 Berkeley Tech. L.J. 443, 444–45 (2005). In the early days of mainframes, before the advent of personal computers, basic source code was usually freely provided with these machines. *Id.* This enabled Stallman to cleverly modify the source code to a network printer, prone to paper jams, so that other users would be alerted when the printer required attention. *Id.* As time went on, however, hardware became more sophisticated and the resulting software more complex. *Id.* Hardware companies began to rely on separate software companies. *Id.* As a result, when MIT's Artificial Intelligence lab received a new Xerox printer, a nondisclosure agreement with Xerox prevented Stallman from accessing the software to implement similar fixes. *Id.*; *see also* Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 Tex. L. Rev. 961, 968–69 (2005).

[17] Carver, *supra* note 16, at 446.

[18] The Unix System: History and Timeline, http://www.unix.org/what_is_unix/history_timeline.html (last visited May 20, 2008). Unix, a simple and elegant operating system developed in the early 1970's by Bell Laboratories, was the standard at the time. *Id.*

[19] GNU Operating System: GNU is Free Software, http://www.gnu.org/ (last visited May 20, 2008).

[20] Free Software Foundation, http://www.fsf.org/ (last visited May 20, 2008). The FSF was established in 1985 with the goal of promoting and developing free software and advocating the software user's rights to study, copy, modify, and redistribute that software. *Id.*

[21] Robert W. Gomulkiewicz, *General Public License 3.0: Hacking the Free Software Movement's Constitution*, 42 Hous. L. Rev. 1015, 1019 (2005). Free software, in addition to being open source, is also essentially free of cost. *Id.* The user-licensee pays no more than a small charge to reproduce the software on a given media. *Id.* The Open Source Initiative ("OSI") is the primary representative of the larger open source community. Open Source Initiative, http://www.opensource.org/docs/osd (last visited May 20, 2008). It defines ten factors of compliance that must be met for software to be considered open source: (1) free distribution; (2) source code must be included or easily obtainable; (3) derived works must be allowed under the same terms as the original license; (4) integrity of the author's source code; (5) no discrimination against persons or groups; (6) no discrimination against fields of endeavor; (7) distribution of license; (8) license must

defined by FSF as: (1) the freedom of a user to run a program for any purpose; (2) the freedom of a user to learn how a program works and modify it to meet his own needs; (3) the freedom of a user to redistribute copies to help his colleagues; and (4) the freedom to improve a program for the benefit of the whole community and release it to the public.[22] In its early years, FSF mainly provided monetary support enabling programmers to develop free software, but today, its employees and volunteers deal largely with the legal issues surrounding free software, most notably licensing.[23]

B. The GNU General Public License

1. First Version

While Stallman created separate licenses for early versions of his GNU software,[24] his goal was a universal license form that could be used to keep all programs free.[25] In 1989, the FSF sought to combat the two major impediments to "software freedom" by releasing the first version of the GNU General Public License ("GPLv1").[26] The license ensured that GPLv1-licensed software was distributed along with its human readable source code and was not integrated with software that would impose additional restrictions on the combination.[27] However, this first version failed to protect Stallman's original GNU Emacs program,[28] thus magnifying the need for extensive revisions to the GPL license.[29]

not be specific to a product; (9) license must not restrict other software; and (10) license must be technology-neutral. *Id.*; *see* Raymond T. Nimmer, *Legal Issues in Open Source and Free Software Distribution*, *in* OPEN SOURCE SOFTWARE FALL 2006: CRITICAL ISSUES IN TODAY'S CORPORATE ENVIRONMENT 2006, at 33, 43–48 (PLI Pats., Copyrights, Trademarks, & Literary Prop., Course Handbook Ser. No. 885, 2006), *available at* WL, 885 PLI/Pat 33.

[22] GNU Project: The Free Software Definition, http://www.gnu.org/philosophy/free-sw.html (last visited May 20, 2008). FSF maintains this definition on their website to clearly show what they feel is critical for software to be free. *Id.*

[23] Free Software Foundation: About Us, http://www.fsf.org/about (last visited May 20, 2008). Since more and more developers have been authoring free software since the mid-1990's, their interests are best served if FSF focuses on the legal issues surrounding free software. *Id.*

[24] Dennis M. Kennedy, *A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture*, 20 ST. LOUIS U. PUB. L. REV. 345, 350–51 (2001).

[25] Gomulkiewicz, *supra* note 21, at 1024. GNU software such as GNU Emacs, GNU Debugger, and GNU Compiler Collection is maintained and used to seed the free software movement. *Id.*

[26] Free Software Foundation: FSF Releases the GNU General Public License, Version 3, http://www.fsf.org/news/gplv3_launched (last visited May 20, 2008); Gomulkiewicz, *supra* note 21, at 1026.

[27] GNU Project: GNU General Public License Version 1, http://www.gnu.org/licenses/gpl-1.0.txt (last visited Apr. 25, 2008).

[28] Gomulkiewicz, *supra* note 21, at 1025–26. Stallman incorporated Emacs source code into his own Emacs program. *Id.* The Emacs source code was originally written by James Gosling, the creator of Java, who reportedly allowed free distribution of the source code. *Id.* When Gosling later sold the rights in Emacs to UniPress, Stallman was forced to rewrite the relevant portion of his program. *Id.*

[29] *See id.* at 1026.

2. *Second Version*

GNU GPL version 2 ("GPLv2") is based on a unique idea that the FSF calls "copyleft."[30] Whereas a copyright is used to restrict use of copyrighted material, copyleft is used to promote it.[31] Copyleft allows an end user to modify and redistribute associated software, but also requires that the modified copy, along with its source code, be distributed under the GPL.[32] If these terms cannot be met, then redistribution is strictly forbidden.[33] This gives the GPL a "viral" character in the sense that if protected code is "injected" into another program, it "infects" that program, effectively binding the combined product to the GPL agreement as well.[34]

[30] GNU Project: What is Copyleft?, http://www.gnu.org/copyleft/copyleft.html (last visited Oct. 3, 2007). The Free Software Foundation defines "copyleft" in the following manner:

> *Copyleft* is a general method for making a program free software and requiring all modified and extended versions of the program to be free software as well.
>
> The simplest way to make a program free is to put it in the public domain, uncopyrighted. This allows people to share the program and their improvements, if they are so minded. But it also allows uncooperative people to convert the program into proprietary software. They can make changes, many or few, and distribute the results as a proprietary product. People who receive the program in that modified form do not have the freedom that the original author gave them; the middleman has stripped it away.
>
> In the GNU project, our aim is to give *all* users the freedom to redistribute and change GNU software. If middlemen could strip off the freedom, we might have many users, but those users would not have freedom. So instead of putting GNU software in the public domain, we "copyleft" it. Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. Copyleft guarantees that every user has freedom.

Id.

[31] *Id.*

[32] Mitchel L. Stoltz, *The Penguin Paradox: How the Scope of Derivative Works in Copyright Affects the Effectiveness of the GNU GPL*, 85 B.U. L. REV. 1439, 1441 (2005); Lori E. Lesser, *A Hard Look at the Tough Issues in Open Source Licenses*, *in* OPEN SOURCE SOFTWARE 2005: CRITICAL ISSUES IN TODAY'S CORPORATE ENVIRONMENT 2005, at 7, 21 (PLI Pats., Copyrights, Trademarks, & Literary Prop., Course Handbook Ser. No. 846, 2006), *available at* WL, 846 PLI/Pat 7.

[33] Stoltz, *supra* note 32, at 1441.

[34] Nimmer, *supra* note 21, at 44; Jyh-An Lee, *New Perspectives on Public Goods Production: Policy Implications of Open Source Software*, 9 VAND. J. ENT. & TECH. L. 45, 51–52 (2006). This issue is greatly complicated by the question of what constitutes a derivative work when used in reference to software. Stoltz, *supra* note 32, at 1448–53. This is because there are different ways for software to be "linked" to the protected code. *Id.* It may be the case that the protected code is not incorporated directly into the parent code, but rather that it is used in a subroutine. *Id.* A subroutine is a separate code module that is tasked with a particular job. *Id.* Further difficulty arises in that the parent code can link with the module either statically or dynamically. *Id.* Static linking occurs when the module is compiled together with the parent code into one object code. *Id.* In dynamic linking, the module is compiled separately resulting in two discrete programs that then interface with each other and exchange data during run time. *Id.*

3. Linux

Linux is the most well-known operating system to be distributed under GPL.[35] Linus Torvalds developed the original kernel[36] while a graduate student at the University of Helsinki.[37] The Linux operating system is also the most well-known example of open source software.[38] Contributors may freely build upon the work of their predecessors, enabling a global community of companies and enthusiasts to continuously improve the operating system since 1991.[39] Linux now has millions of users and arguably surpasses the Windows operating system[40] in customizability, reliability, and power.[41]

Because the GPL provisions forbid charging more for Linux than its cost of distribution, vendors have created a service market.[42] While the source code is freely provided to all users, technical support, services, proprietary programs, and in some cases indemnification[43] are sold for profit to those willing to pay for them.[44]

[35] Kenneth J. Rodriguez, *Closing the Door on Open Source: Can the General Public License Save Linux and Other Open Source Software*, 5 J. HIGH TECH. L. 403, 404–05 (2005); Stoltz, *supra* note 32, at 1472.

[36] M. Tim Jones, Consultant Engineer, Emulex Corp., Anatomy of the Linux Kernel: History and Architectural Decomposition (June 6, 2007), http://www.ibm.com/developerworks/linux/library/l-linux-kernel/?S_TACT=105AGX59&S_CMP=GR&ca=dgr-lnxw01LKernalAnatomy. The kernel is a program that constitutes the central core of an operating system which is responsible for maintaining control over everything that occurs in the system. *Id.*; *see also* Stoltz, *supra* note 32, at 1448.

[37] Linux Online: Linus Torvalds Bio, http://www.linux.org/info/linus.html (last visited May 20, 2008).

[38] Dr. David Crooke, CTO, Convio, Ask the Expert: What's the Difference Between Open Source, Open Standards, and Open APIs? (Oct. 2007), http://www.convio.com/resources/newsletter/opensource-openstandards-openapi.html.

[39] Ibiblio Linux Archive: Ibiblio's Mission Statement, http://www.ibiblio.org/pub/linux/POLICY.html (last visited May 20, 2008); Rodriguez, *supra* note 35, at 405.

[40] Microsoft Windows: compare editions, http://www.microsoft.com/windows/products/windowsvista/editions/choose.mspx (last visited May 20, 2008). Microsoft Windows is a proprietary operating system developed and marketed by Microsoft. *Id.*

[41] Andrew LaFontaine, *Adventures in Software Licensing:* SCO v. IBM *and the Future of the Open Source Model*, 4 J. TELECOMM. & HIGH TECH. L. 449, 458 n.37 (2006); Peter Gali, *Windows Server 2008 Features Address Linux Challenge: Keeping Up With Linux*, EWEEK, May 17, 2007, http://www.eweek.com/c/a/Windows/Windows-Server-2008-Features-Address-Linux-Challenge/1/; Nimmer, *supra* note 21, at 43, 46; Rodriguez, *supra* note 35, at 404–05.

[42] David S. Evans & Anne Layne-Farrar, *Software Patents and Open Source: The Battle over Intellectual Property Rights*, 9 VA. J. L. & TECH. 10, ¶ 18 (2004); Carver, *supra* note 16, at 456; Lesser, *supra* note 32, at 16–17.

[43] Vetter, *supra* note 4, at 209. Corporate users often rely on indemnity clauses to shield them from liability should infringement occur. *Id.* at 208. It is not unusual for traditional software distributors to offer this sort of indemnity. *Id.* As a larger percentage of the corporate world is now turning to free and open source software, there is an increased expectation of indemnity. *See* Martin LaMonica, *Insurer Launches $10 Million Open-Source Policy*, CNET NEWS, Oct. 31, 2005, http://www.news.com/2100-7344-5924112.html (stating that Lloyd's of London will underwrite a policy indemnifying corporate open source customers); *see also* Lee, *supra* note 34, at 108 (stating that major vendors are beginning to offer protection for customers against patent-infringement suits).

[44] Evans & Layne-Farrar, *supra* note 42, ¶ 18; Lesser, *supra* note 32, at 16–17.

C. Litigation

1. History of Software Patents

Although the U.S. Supreme Court decided three landmark software patent cases from 1972 to 1981,[45] it struggled with the issue of software's patentability.[46] Finally, in *Diamond v. Diehr*,[47] the Court tipped the balance in favor of patentability and sparked a trend that would see an increasing variety of software becoming patentable.[48] In response to the First Circuit ruling in *Lotus Development Corp. v. Borland International*[49] that a spreadsheet program was a "method of operation" and therefore not copyrightable subject matter,[50] companies increasingly turned to patent protection for their software.[51]

2. The SCO Group Initiates a Lawsuit Frenzy

Problems can occur when a licensee incorporates proprietary or patented software into GPL software and redistributes it to the public.[52] The SCO Group ("SCO")[53] made this claim in a $1 billion lawsuit filed against IBM on March 6, 2003.[54] SCO alleged that IBM gleaned and then incorporated SCO's intellectual property into the Linux project during a failed collaboration attempt to produce an advanced version of Unix for Intel's new Itanium processor.[55]

Responding to the threat against Linux, two more open source providers jumped into the fray. On May 28, 2003, Novell Corp. ("Novell") publicly challenged SCO's allegations by announcing that it never sold the Unix copyrights to Santa Cruz Operation,[56] SCO's predecessor, as part of an earlier asset purchase agreement.[57]

[45] Diamond v. Diehr, 450 U.S. 175, 193 (1981) (holding that computer programs that are but one part of a useful process are patentable); Parker v. Flook, 437 U.S. 584, 596 (1978) (holding that a method for simply computing a number is not patentable); Gottschalk v. Benson, 409 U.S. 63, 73 (1972) (holding that a software algorithm is nothing more than a mathematical recipe to be run on a computer and therefore not patentable).

[46] JANICE M. MUELLER, AN INTRODUCTION TO PATENT LAW 218–20 (Aspen Publishers 2d ed. 2006).

[47] 450 U.S. 175.

[48] *Id.* 192–93; Grant C. Yang, *The Continuing Debate of Software Patents and the Open Source Movement*, 13 TEX. INTELL. PROP. L. J. 171, 178 (2005); Evans & Layne-Farrar, *supra* note 42, ¶¶ 9–11.

[49] 49 F.3d 807 (1st Cir. 1995).

[50] *Id.* at 819 (holding that even though the defendant was found to have copied plaintiff's menu command, that command was not copyrightable subject matter because it was a method of operation).

[51] Mann, *supra* note 16, at 972.

[52] Evans & Layne-Farrar, *supra* note 42, ¶ 20.

[53] SCO Group, http://www.thescogroup.com/ (last visited May 20, 2008).

[54] Amended Complaint, SCO Group, Inc. v. IBM, No. 03-CV-0294 (D. Utah July 22, 2003); *see also* Carver, *supra* note 16, at 473.

[55] Amended Complaint, *supra* note 54, ¶ 89; Carver, *supra* note 16, at 473.

[56] *See* The History of the SCO Group, http://www.thescogroup.com/company/history.html (last visited May 20, 2008). The Santa Cruz Operation (not to be confused with the SCO Group) was a California software company that sold Xenix, SCO UNIX, and UnixWare—all variants of Unix. *Id.* SCO merged with two smaller companies in 1993 and picked up a new product line called

Novell then defiantly registered several key Unix copyrights.[58] On August 4 of the same year, Red Hat Corp. ("Red Hat") filed suit against SCO seeking an injunction and declaratory judgment stating that Red Had had not infringed SCO's copyrights.[59]

SCO responded to Novell on January 20, 2004, by filing a Slander of Title lawsuit against Novell.[60] On March 3, 2004, SCO filed suit against Daimler Chrysler and AutoZone because they failed to respond to a demand letter for license fees sent out by SCO to Unix users licensed under Novell.[61]

Many analysts surmised that the initiation of SCO's lawsuit against IBM was nothing more than a ploy to create publicity and boost SCO stock.[62] Judge Brook Wells, magistrate judge of the U.S. District Court of Utah, even remarked that SCO's tactics were akin to those of an officer refusing to disclose what an arrested shoplifter is accused of stealing.[63] It was the August 10, 2007, ruling in the *SCO-Novell* case that put an end to this web of litigation.[64] However, the integrity of the GPL was never tested because Judge Kimball held that Novell, not SCO, was the rightful owner of the copyrights covering Unix.[65]

Tarantella. *Id.* SCO then shed their Unix product lines by selling the rights to Caldera Systems, and changed its name to Tarantella, Inc. *Id.* Caldera then later changed its name to The SCO Group. *Id.*

[57] SCO Group, Inc. v. Novell, Inc., 377 F. Supp. 2d 1145, 1147 (D. Utah 2005). SCO claimed there was a second amendment to the asset purchase agreement between Novell and Santa Cruz Operation that supports SCO's assertion that SCO and not Novell acquired the rights to Unix in question. *Id.* In its public statement Novell pointed out that it had applied for and received copyright registrations covering the portions of Unix in question. *Id.* In its motion to dismiss, Novell argued SCO's purchase agreement by itself does not prove a valid transfer of copyright ownership. *Id.*

[58] *Id.* at 1149.

[59] *See* Red Hat, Inc. v. SCO Group, Inc., No. 03-772-SLR, 2004 U.S. Dist. LEXIS 7077, at *5 (D. Del. Apr. 6, 2004) (ruling that action should be stayed pending the outcome of the litigation between SCO and IBM).

[60] Complaint, SCO Group, Inc. v. Novell, Inc., 377 F. Supp. 2d 1145 (D. Utah 2005) (No. 2:04CV139DAK).

[61] SCO Group, Inc. v. DaimlerChrysler Corp., No. 04-056587-CKB (Mich. Cir. Ct. filed Mar. 3, 2004); SCO Group, Inc. v. AutoZone, Inc., No. CV-S-04-0237-DWH-LRL (D. Nev. Filed Mar. 3, 2004); *see* Lori E. Lesser, *Open Source Software 2006: Critical Issues in Today's Corporate Environment*, *in* OPEN SOURCE SOFTWARE FALL 2006: CRITICAL ISSUES IN TODAY'S CORPORATE ENVIRONMENT 2006, at 9, 28 (PLI Pats., Copyrights, Trademarks, & Literary Prop., Course Handbook Ser. No. 885, 2006), *available at* WL, 885 PLI/Pat 9; SCO Group: *SCO v. Daimler Chrysler*, http://www.sco.com/scoip/lawsuits/daimlerchrysler/index.html (last visited Apr. 19, 2008); SCO Group: *SCO v. AutoZone*, http://www.sco.com/scoip/lawsuits/autozone/index.html (last visited Apr. 19, 2008).

[62] Rodriguez, *supra* note 35, at 410–11.

[63] Order Granting in Part IBM's Motion to Limit SCO's Claims at 34, SCO Group, Inc. v. IBM, No. 2:03cv00294DAK (D. Utah June 28, 2006).

[64] SCO Group, Inc. v. Novell, Inc., No. 2:04CV139DAK, 2007 U.S. Dist. LEXIS 58854, at *158 (D. Utah Aug. 10, 2007) (holding that Novell is the rightful owner of Unix and UnixWare copyrights).

[65] *Id.* The SCO Group meant to challenge the very legal foundation on which GPL was built. Rodriguez, *supra* note 35, at 415. SCO's CEO proclaimed that the entire legal model for U.S. copyright law rests on the idea of competition and profit motive. *Id.* These questions would have been addressed had the court held that SCO owned the copyrights to Unix. *Id.* at 409. On September 14, 2007, SCO announced its filing for a petition for reorganization under Chapter 11 of the U.S. Bankruptcy Code to protect against future financial and legal treats to its assets. *SCO, a*

3. Off Track

In 2007, another type of open source license[66] was tested in court. A group of model railroad enthusiasts[67] developed source code that they copyrighted under their own open source artistic license.[68] The KAM software company,[69] based in Portland, Oregon, secured patents on software that contained this code and thereafter sought to collect royalties.[70] The court ruled that "possibly commercial" uses by the licensee were implicit in the phrasing of the license.[71]

D. Defense Against Patents

1. The New, Beefier GNU GPL Version 3.0

GNU GPL version 3 ("GPLv3") was released by the FSF on June 29, 2007.[72] With regard to patents, it enhances the protection of free software in two ways. First, GPLv3 makes explicit the implicit patent grant clause of GPLv2.[73] Second, GPLv3 contains a cross-licensing restriction clause.[74] If a distributor of GPL-licensed software holds a patent claim on code that is incorporated into that software, then by the mere act of distribution the distributor/patentee extends a patent license to all downstream recipients.[75]

Patent cross-licensing occurs when individual patent holders agree to share their protected intellectual property.[76] Complications arise when one party incorporates the other party's protected code into GPL-licensed software and then

Software Maker, Is Seeking Bankruptcy, N.Y. TIMES, Sept. 15, 2007, at C4; SCO Group: SCO Files for Chapter 11 Protection, Sep. 14, 2007, http://www.thescogroup.com/chapter_11/.

[66] Open Source Initiative: Open Source Licenses (Sept. 19, 2006), http://www.opensource.org/licenses. The Open Source Initiative lists sixty different open source licenses on their website. *Id.*

[67] JMRI Defense: Keeping an Open Source Project Alive, http://jmri.sourceforge.net/k/index.html (expressing their views on their lawsuit) (last visited May 20, 2008). The Java Model Railroad Interface ("JMRI") Project is an online community that develops open source model-trail software. *Id.*

[68] Jacobsen v. Katzer, No. C06-01905JSW, 2007 U.S. Dist. LEXIS 63568, at *17–18 (D. Cal. Aug. 17, 2007).

[69] About KAM Industries, http://www.trainpriority.com/Aboutus/AboutKam.aspx (last visited May 20, 2008).

[70] Jacobsen, 2007 LEXIS 63568, at *2.

[71] Jacobsen, 2007 LEXIS 63568, at *18–19.

[72] Free Software Foundation: FSF Releases the GNU General Public License Version 3, http://www.fsf.org/news/gplv3_launched (last visited Oct. 27, 2007). The release of GNU version 3.0 follows a nearly two-year process where the public could weigh in with suggestions and concerns. *Id.*

[73] Richard Stallman, *Why Upgrade to GPLv3*, Nov. 6, 2007, http://www.gnu.org/licenses/rms-why-gplv3.html.

[74] *Id.*

[75] Shaobin Zhu, *Patent Rights Under FOSS Licensing Schemes*, 4 SHIDLER J.L. COM. & TECH. 4, paras. 19–20 (2007), http://www.lctjournal.washington.edu/Vol4/a04zhu.html.

[76] Mann, *supra* note 16, at 990; Nimmer, *supra* note 21, at 101.

distributes it downstream to third-party users.[77] The scenario is not surprising in light of submarine patents[78] and the growing patent thicket.[79]

2. The Open Invention Network

The Open Invention Network ("OIN")[80] is an intellectual property company that hopes to leverage a patent pool[81] against the threat of suits against Linux.[82] The patents OIN acquires through purchases and donation are licensed royalty-free to companies if they affirmatively agree not to assert their own patent rights against Linux software.[83] The concept of patent pooling has been legally challenged on unfair competition grounds.[84] In August of 2007, Linux-driven search engine giant Google became the thirteenth member of OIN.[85]

[77] Nimmer, *supra* note 21, at 100.

[78] Evans & Layne-Farrar, *supra* note 42, ¶ 37. Hidden or submarine patents can be troublesome for software developers. *Id.* A company may release a computing format in hopes of gaining widespread acceptance as an industry standard. *Id.* All the while, the company is secretly prosecuting the patent. *Id.* When the format is well ensconced, the unsuspecting developers are blindsided by enforcement demands. *Id.* This danger is enhanced by the fact that software cycles so quickly, many generations can pass before a patent issues. Yang, *supra* note 48, at 187. This was the case when the Lempel-Ziv-Welch ("LZW") lossless data compression algorithm was published by one of the inventers in a trade magazine. Evans & Layne-Farrar, *supra* note 42, ¶ 37. The fact that Unisys held the rights to the method and was in the process of obtaining a patent was not disclosed in the article. *Id.* The algorithm even became the basis for the very popular GIF format. *Id.* A full decade after disclosing the LZW method to the public—when the algorithm was firmly entrenched—Unisys asserted its patent rights. *Id.*

[79] David A. Balto, *Intellectual Property and Antitrust: General Principles, in* INTELLECTUAL PROP. ANTITRUST 2007, at 9, 87 (PLI Pats., Copyrights, Trademarks, & Literary Prop., Course Handbook Ser. No. 885, 2007), *available at* WL, 867 PLI/Pat 9. The proliferation of software patents by large companies leads to a "patent thicket" that must be "hacked" through. *Id.* The greater the number of patents, the greater the statistical likelihood that routine development will result in accidental infringement. *Id. See also* Mann, *supra* note 16, at 999.

[80] Open Invention Network, http://www.openinventionnetwork.com/about.php (last visited Oct. 27, 2007). OIN is based in New York City. *Id.* It was founded on November 10, 2005 by IBM, Novell, Red Hat, Philips, and Sony. Ryan Paul, *IBM, Sony, Novell Start Linux Patent Sharing Project*, ARS TECHNICA, Nov. 10, 2005, http://arstechnica.com/news.ars/post/20051110-5553.html.

[81] Harvey I. Saferstein, *Patent Licensing and the Antitrust Laws, in* PATENT LAW INSTITUTE 2007: THE IMPACT OF RECENT DEVELOPMENTS ON YOUR PRACTICE 2007, at 913, 959–60 (PLI Pats., Copyrights, Trademarks, & Literary Prop., Course Handbook Ser. No. 885, 2007), *available at* WL, 899 PLI/Pat 913.

[82] Open Invention Network, http://www.openinventionnetwork.com/about.php (last visited May 20, 2008).

[83] *Id.*

[84] Wallace v. IBM, 467 F.3d 1104, 1106 (7th Cir. 2006) (affirming the district court's dismissal of the plaintiff's complaint). Wallace wanted to write his own operating system for profit but felt he could not compete with Linux, which is available for free. *Id.* He contended that defendants IBM, Red Hat, and Novell have conspired with FSF and others by pooling and cross-licensing their patents. *Id.* Wallace labels this practice as a predatory price-fixing scheme and a violation of antitrust laws. *Id.* The U.S. Court of Appeals for the Seventh Circuit held that predatory pricing amounts to raising prices after the competition has exited. *Id.* In this case the price of Linux remains fixed at zero. *Id.* The court further held that Wallace has no claims under antitrust law because they are meant to protect the interests of the consumer, not the producer. *Id.* Wallace claimed in a separate suit against FSF that it unlawfully conspired with distributors of Linux to fix

3. Prior Art Database

The U.S. Patent and Trademark Office ("USPTO") is now beginning to collaborate with open software developers in an effort to improve the quality of software patents.[86] The USPTO weathered much criticism from open source advocates who contend that the rapid issuance of patents without proper investigation leads to a flurry of infringement lawsuits.[87] IBM proposes to provide the USPTO with a massive automated database containing subcategories for open source software.[88] By promptly adding new open source developments to the existing database, developers create a pool of prior art that disqualifies similar or identical material from receiving a patent.[89]

II. ANALYSIS

The notion of driving innovation by offering a time-limited monopoly is written into the U.S. Constitution.[90] However, given that mere mathematical algorithms are unpatentable,[91] courts struggled with the patentability of computer programs.[92] The open source community argues that favorable court decisions regarding software

the price through the use of the GPL. Wallace v. Free Software Found., Inc., No. 1:05-cv-0618-JDT-TAB, 2006 U.S. Dist. LEXIS 53003, at *2–3 (S.D. Ind. Mar. 20, 2006) (reasserting dismissal motion and denying plaintiff leave to further amend his complaint). As a result, he was unable to enter the marketplace. *Id.* The court held he had failed to allege an injury and allowed him to amend his complaint. *Id.* at *12. Because Wallace again failed to assert an injury, his complaint was dismissed without leave to further amend it. *Id.*

[85] Charles Babcock, *Google Joins Effort to Ward Off Patent Challenges to Linux*, INFO. WK., Aug. 13, 2007, at 35.

[86] John Markoff, *U.S. Office Joins an Effort to Improve Software Patents*, N.Y. TIMES, Jan. 10, 2006, C3 (announcing a plan by USPTO to cooperate with open source software developers to create searchable Internet databases that will help both examiners and filers gage the novelty and quality of patent applications).

[87] *Id.*

[88] *Id.*

[89] Carver, *supra* note 16, at 463–64.

[90] U.S. CONST. art. I, § 8, cl. 8. The Patent Act attempts "to promote the Progress of Science and useful Arts by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." *Id.*

[91] *See, e.g.*, Le Roy v. Tatham, 55 U.S. 156, 175 (1852) (holding that a principle in the abstract is a fundamental truth and cannot be patented).

[92] Gottschalk v. Benson, 409 U.S. 63, 73 (1972); Parker v. Flook, 437 U.S. 584, 596 (1978); Diamond v. Diehr, 450 U.S. 175, 193 (1981); *see also* 35 U.S.C. § 102 (2006). Only a new and useful process, machine, manufacture, composition of matter, or a new and useful improvement thereof is eligible for a patent. *Id.*

patents have actually had a chilling effect on software innovation,[93] while Microsoft has declared open source software as "un-American."[94]

A. Moving Away from What Worked

The fact that the noncompetitive open source model has burgeoned into a multi-billion dollar software industry is proof that innovation will continue in the absence of software patents.[95] Linux alone currently enjoys at least ten million users,[96] which includes over ninety-nine different governments in forty-four countries.[97] Indeed, both open source and proprietary software development thrived when copyright law was all that was available and required for software protection.[98]

Beginning in the 1990s, the courts decided a series of cases that began to erode the software copyright to the point where it ceased to provide adequate protection.[99] The courts established this unsettling precedent while concurrently taking an evermore permissive stance on software patents.[100] The First Circuit decision in *Lotus*[101] provided the flashpoint for the massive patent migration to follow.[102]

[93] Mann, *supra* note 16, at 981. The resource required for prosecuting and litigating patents could otherwise be directed toward research and development. *Id.* The strategy of "over-patenting," to gain a stronger bargaining position in cross-licensing negotiations, adds to the problem of the patent anti-commons, which stifles software innovation. Evans & Layne-Farrar, *supra* note 42, ¶ 54. Patents also block open source programmers for exploring certain innovative ideas that they might otherwise have. *Id.* at 22.

[94] Alan Story, *Don't Ignore Copyright, the 'Sleeping Giant' on the TRIPs and International Education Agenda*, in GLOBAL INTELLECTUAL PROPERTY RIGHTS, KNOWLEDGE, ACCESS, AND DEVELOPMENT 125, 135 (Peter Drahos & Ruth Mayne eds., 2002).

[95] Stolz, *supra* note 32, at 1440–41; Zhu, *supra* note 75, para. 1.

[96] Rodriguez, *supra* note 35, at 408 (reporting about 7.5 million Linux users in 2005). Counter.li.org reports an estimated 29 million Linux users in 2005. Linux Counter: Estimating the Number of Linux Users (2005), http://counter.li.org/estimates.php.

[97] Lee, *supra* note 34, at 56.

[98] Jane C. Ginsburg, *Four Reasons and a Paradox: the Manifest Superiority of Copyright Over SUI Generis Protection of Computer Software*, 94 COLUM. L. REV. 2559, 2559 (1994); *see also* Mann, *supra* note 16, at 971 (stating that copyright provided relatively strong protection for software until the late 1980s).

[99] Computer Assocs. Int'l v. Altai, 982 F.2d 693, 705 (2d Cir. 1992) (affirming dismissal of copyright infringement claim holding that defendant's computer program contained no protectable expression). The court criticized an earlier decision that was more liberal in its application of copyright protection and put forth its own rule of "abstraction-filtration comparison." *Id.* The Second Circuit's position in *Altai* became the majority position among all the circuit courts which limited the availability of copyright protection for computer software. Yang, *supra* note 48, at 176–77.

[100] *In re* Alappat, 33 F.3d 1526, 1537 (Fed. Cir. 1994) (reversing the USPTO's patent invalidation of "a means for creating a smooth waveform display in a digital oscilloscope"). This ruling fixed the statutory standing on software patents. Evans & Layne-Farrar, *supra* note 42, at ¶ 10. In 1998, the Federal Circuit ruled that a machine or program that stores and calculates numbers in a useful manner is eligible for a patent. State St. Bank & Trust Co. v. Signature Fin. Group, 149 F.3d 1368, 1373 (Fed. Cir. 1998).

[101] Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 819 (1st Cir. 1995).

[102] *See* L. Donald Prutzman, *United States Patent Protection for Computer Software*, 19 INT'L L. PRACTICUM 45, 49 (2006) (stating that the significantly weakened copyright protection for

Prolific proprietary software companies congested the software industry with patents, and while they provided relatively weak economic returns, the patents were put to uses that Congress had never intended. Cross-licensing is such a use and is regarded by many as being exceptionally damaging to software innovation.[103]

B. Hacking Through the Thicket

Today's programs are complex.[104] Each new generation of software builds upon the last.[105] Without the need to reinvent the wheel, programmers typically patch hundreds or even thousands of pre-existing programs and algorithms together in such a way as to produce a novel result.[106] The originality of developers' contributions is usually quite small.[107]

The proliferation of software patents has created an "anticommons" by which large numbers of "building block" programs have become legally inaccessible.[108] The programmer must navigate this "thicket" by designing around the forbidden components.[109] This slows forward progress by diverting resources away from innovation.

Numerous additional patents, which others must also design around, are required to effectively protect the ideas behind new programs. This "catch-22" scenario succeeds only in making an already troubling patent thicket even denser.

C. An Undeserved Reward

The right to exclude others becomes important when research and development costs are high.[110] Pharmaceutical companies, for example, must invest substantial amounts of time and money in developing effective and beneficial drugs.[111] Therefore, patents become essential in allowing such companies the opportunity to recoup their investment and harvest the fruits of their labor by excluding others from

software in the wake of *Lotus* served as an additional impetus for the software patenting movement).

[103] Webbink, *supra* note 1, ¶¶ 14–19.

[104] Evans & Layne-Farrar, *supra* note 42, ¶ 53. Theoretical challenges facing modern day programmers are becoming increasingly more complex. *Id.*

[105] *Id.* Pre-existing pieces of code can be incorporated into new programs to perform routine functions. *Id.*

[106] *Id.*

[107] *Id.* Novelty is not limited to original code and is also inherent in the way preexisting code is assembled. *Id.*

[108] Mann, *supra* note 16, at 999.

[109] Evans & Layne-Farrar, *supra* note 42, ¶ 54.

[110] ROGER SCHECHTER & JOHN THOMAS, PRINCIPLES OF PATENT LAW 12 (2004). If companies were not given the opportunity to recoup their research and development costs on expensive endeavors, then they would not commit to those ventures. *Id.*

[111] *Post-Approval R&D Raises Total Drug Development Costs to $897 Million,* TUFTS CENTER FOR THE STUDY OF DRUG DEVELOPMENT IMPACT REPORT, May/June 2003. The average cost of developing a new prescription drug in 2003 was $897 million, four times the figure in the early 1990's. *Id.*

a "free ride."[112] Conversely, the cost of software development is relatively low, with turnaround times typically measured in months.[113] Granting twenty years of exclusionary rights in such circumstances is an unjust reward.

D. Veiled Infringement

The use and exploitation of protected material in large software projects is inevitable due to the patent thicket.[114] In 2004, Open Source Risk Management ("OSRM"),[115] a leading code auditing firm, reported that 283 patents had been incorporated into the Linux kernel, of which only a third were held by Linux-friendly companies.[116] The fact that the Linux kernel was comprised of approximately six-million lines of code in 2004 helps put this statistic into perspective.[117] Because open source code is distributed in human-readable form, it is left vulnerable to infringement claims.[118] Programmers must be vigilant in guarding against the use of patented code and any subsequent litigation threat.[119]

There is wide speculation that proprietary software companies take advantage of freely available open source code by incorporating it into their own programs.[120] Because such companies keep their source code confidential and distribute only the proprietary object code, they effectively hide behind a virtual veil of obscurity that makes assessments of infringement nearly impossible.[121]

[112] Yang, *supra* note 48, at 193. A "free-ride" is where one company utilizes the unprotected results from another companies research and development efforts. *Id.* In this way a company can avoid the costs associated with developing a technology on its own. *Id.*

[113] Yang, *supra* note 48, at 187–88; Nimmer, *supra* note 21, at 46; Lee, *supra* note 34, at 50; Mann, *supra* note 16, at 979; Vetter, *supra* note 4, at 205.

[114] Evans & Layne-Farrar, *supra* note 42, at *54; Mann, *supra* note 16, at 999; *see also* Carver, *supra* note 16, at 460.

[115] Open Source Risk Management: About OSRM, http://www.osriskmanagement.com/about.html (last visited May 20, 2008). OSRM is an open source software risk management consulting firm founded in 2003. *Id.* They audit code to help identify issues that could impact licensing issues. *Id.*

[116] Press Release: Open Source Risk Management, *Results of First-Ever Linux Patent Review Announced, Patent Insurance Offered by Open Source Risk Management*, http://www.osriskmanagement.com/press_releases/press_release_080204.pdf (last visited Apr. 25, 2008).

[117] IBM: Anatomy of the Linux Kernel, http://www.ibm.com/developerworks/linux/library/l-linux-kernel/?S_TACT=105AGX59&S_CMP=GR&ca=dgr-lnxw01LKernalAnatomy (last visited May 20, 2008). The Linux kernel grew from roughly 10,000 lines of code in 1991, to over 6 million lines in 2004. *Id.*

[118] Lee, *supra* note 34, at 109. Under the terms of its license, human-readable source code must be distributed with open source software. Nimmer, *supra* note 21, at 46.

[119] Lee, *supra* note 34, at 109. Proprietary code finding its way into open source software is compounded by the fact that many individuals may have had a hand in developing the code. Carver, *supra* note 16, at 460. As programs are passed from one developer to the next, each is focused on his or her own contributions and often accepts on faith that the software was free of proprietary source when received. Christian H. Nadan, *Open Source Licensing: Virus or Virtue?*, 10 TEX. INTELL. PROP. L.J. 349, 352 (2002).

[120] Lee, *supra* note 34, at 109; *see also* Carver, *supra* note 16, at 460.

[121] Carver, *supra* note 16, at 460. Proprietary source code is first compiled into object code before being distributed. Nadan, *supra* note 119, at 350–51. The object code is the form that is executed by computers and is not readable by humans. *Id.* To make an assessment of infringement

E. Time for Change

Innovation, development, and marketing in the software industry occur at a breakneck pace.[122] It is not uncommon for software to cycle through several iterations on an annual basis.[123] Even with the recent USPTO reform requiring patent applications to be published eighteen months after filing,[124] an entire generation of software may cycle during this period,[125] and the software may well be obsolete before a patent issues.[126] The speed at which open source software evolves is even greater than that of proprietary software.[127] For example, in the five years it took Microsoft to supersede Windows XP with Vista, SUSE Linux, currently owned by Novell, cycled through eleven major releases.[128] Also, the cores of Linux-based operating systems are typically updated several times a month.[129] This rapid evolution leaves software developers particularly vulnerable to the submarine patent.[130]

Contrast this concern with the previous example of bringing a new drug to market.[131] Research and development on new drugs typically takes many years and is followed by a waiting period for FDA approval.[132] The extremely short timeframe for software development, in conjunction with its low development costs, suggests that the software industry might be better served if software patents had an adjusted duration of less than the current twenty-year term.

F. Setting Industry Standards

by proprietary software, it becomes necessary to acquire the proprietary source code which is usually kept very secure. Carver, *supra* note 16, at 460.

[122] Yang, *supra* note 48, at 187; Vetter, *supra* note 4, at 205; Mann, *supra* note 16, at 979; Nimmer, *supra* note 21, at 46; Lee, *supra* note 34, at 50.

[123] *See, e.g.*, EA Tiburon, http://www.tiburon.com/ (last visited May 20, 2008). Madden NFL, developed by Electronic Arts Tiburon for EA Sports, is a video game featuring American football with new additions being released annually. *Id.* Also, the Free BSD operating system is an example of software that goes through multiple annual releases. Clemente Izurieta & James Bieman, *The Evolution of FreeBSD and Linux*, Sept. 21–22, 2006, http://portal.acm.org/portal.cfm (search "The Evolution of FreeBSD and Linux").

[124] 35 U.S.C. § 122(b)(1)(A)(2006). Patent applications are to be published promptly after eighteen months from filing. *Id.*; *see also* Evans & Layne-Farrar, *supra* note 42, ¶ 42.

[125] Yang, *supra* note 48, at 187.

[126] Mann, *supra* note 16, at 979.

[127] Nimmer, *supra* note 21, at 46. Because people continually improve, adapt, and fix bugs, open source software can evolve at a speed that seems astonishing as compared to the slow pace of proprietary software. *Id.*; *see also* Vetter, *supra* note 4, at 205.

[128] Suse Linux: Distributions, http://www.linux.com/distributions/114377 (last visited May 20, 2008). Ubuntu issues a new server and desktop release of Linux every six months. Ubuntu: What is Ubuntu?, http://www.ubuntu.com/products/WhatIsUbuntu (last visited May 20, 2008).

[129] Matthew D. Satchwell, *The Tao of Open Source: Minimum Action for Maximum Gain*, 20 BERKELEY TECH. L.J. 1757, 1775 n.49 (2005). The Fedora project, a popular flavor of Linux, even posts incremental updates online each evening. *Id.*

[130] Yang, *supra* note 48, at 187; Evans & Layne-Farrar, *supra* note 42, ¶ 37.

[131] *See* Yang, *supra* note 48, at 193.

[132] FDA's Drug Review and Approval Times, http://www.fda.gov/cder/reports/reviewtimes/default.htm (last visited Oct. 31, 2007). The FDA approval time for new drugs is a little over a year. *Id.*

While it is true that software diversity provides the luxury of choice to developers and end users alike, it also adds to the problem of compatibility. Industry standards must be enacted if individual software applications are to be able to interact, share data, and perform common tasks.[133] The TCP/IP Internet protocol, for example, is a result of government sponsorship of non-proprietary Internet architecture inventions.[134] The government recognized that as the Internet matured it must be freely available and not solely under the dominion of a single controlling entity.[135] The result is that the Internet today is accessible to all, regardless of the computer brand or operating system used.[136] It is important that such widespread technological standards not be held hostage by any single company owning the relevant patents.[137] The open source community, in contrast, is able to provide open industry standards because it is not driven by economic incentives.[138] Open source developers are even able to produce inventions that would not otherwise have been commercially available because the concepts may not have seemed economically feasible at the time.[139]

G. Skewed Uses for Software Patents

Economically, larger software firms gain little from patents.[140] The opportunity to secure innovation through exclusionary rights alone does not justify the

[133] Evans & Layne-Farrar, *supra* note 42, at ¶ 36. Various hardware and software components, like video cards, hard drives, and file systems, all need to communicate with each other. *Id.* This can only be done if the different software options driving such devices all utilize a uniform, industry wide standard. *Id.*

[134] Lee, *supra* note 34, at 77. It was the United States Department of Defense that was responsible for the world's first operational packet switching network, called ARPAnet. Computer History: Internet History, http://www.computerhistory.org/internet_history/ (last visited May 20, 2008). On January 1, 1983, all networks attached to the ARPAnet adopt the TCP/IP networking protocol and the internet as we know it is born. History of the Internet: Chapter 4, http://www.historyoftheinternet.com/chap4.html (last visited May 20, 2008).

[135] Philip J. Weiser, *The Internet, Innovation, and Intellectual Property Policy*, 103 COLUM. L. REV. 534, 543 (2003). One point of contention is whether continued Congressional action is required to ensure access to the Internet remains unfettered. ANGELE A. GILROY, CONG. RESEARCH SERV., LIBRARY OF CONG., NET NEUTRALITY BACKGROUND AND ISSUES 1, 1 (2007), http://www.fas.org/sgp/crs/misc/RS22444.pdf.

[136] Dan L. Burk, *Law as a Network Standard*, 8 YALE J. L. & TECH. 63, 72 (2006).

[137] Evans & Layne-Farrar, *supra* note 42, ¶ 36; Yang, *supra* note 48, at 203; Zhu, *supra* note 75, para. 3.

[138] Ronald J. Mann, *Commercializing Open Source Software: Do Property Rights Still Matter?*, 20 HARV. J.L. & TECH. 1, 8–9 (2006) [hereinafter Mann—*Commercializing*]. An industry standard developed by the open source community has the additional perk of being freely available to everyone. Stephen Lindholm, *Marking the Software Patent Beast*, 10 Stan. J.L. BUS. & FIN. 82, 127 (2005).

[139] Yang, *supra* note 48, at 203. Since capitalism drives proprietary software companies, would-be software that is judged unprofitable is never developed. *Id.* Programmers code open source software for a variety of different reasons, including a sense of community, education, to build a reputation, or just as a hobby. Lee, *supra* note 34, at 55. Anyone of these reasons may serve as the genesis for bringing an otherwise unprofitable program into existence and distributing it under an open source license. *Id.*

[140] Mann, *supra* note 16, at 996.

prosecution and litigation costs incurred by such entities.[141] Yet strong alternative incentives remain for such firms to seek out a large portfolio of patents.[142] The use of these patents, however, runs contrary to the goal of the Patent Act.[143]

1. Signaling

Large software firms do not seek patents to serve innovation by protecting investments, but rather to signal investors and consumers that they should be recognized as a major technological player.[144] To those seeking acquisition of company shares or products, the message is clear: the company possesses the resources, skill, and competence of leadership to achieve results that are worth patenting.[145] In striving for such standing, these companies will often seek and receive patents that arguably should not have issued.[146]

2. Membership

An additional benefit of a properly padded patent portfolio is the ability to cross-license patents with other large software firms.[147] Large software companies enter into arrangements with other similar entities in which they agree to use each other's exclusive patent rights.[148] These agreements effectively create virtual monopolies that greatly debilitate the efforts of smaller companies and open source developers.[149]

H. Open Source Patent Defenses

[141] *Id.* Since large firms can successfully compete with smaller ones, they gain relatively little from litigation or the exclusion of competitors. *Id.* The cost of continued software innovation is simply far less than implementing legal marketing strategies. Yang, *supra* note 48, at 196.

[142] Mann, *supra* note 16, at 996. Large firms seek the freedom to deploy designs as they wish without restrictions from outside patent holders, and thus build up large cross-licensing portfolios. *Id.*

[143] *See*, Herbert Hovenkamp, Mark Janis, & Mark A. Lemley, *Anticompetitive Settlement of Intellectual Property Disputes,* 87 MINN. L. REV. 1719, 1743 (2003).

[144] Mann, *supra* note 16, at 993–95. When a firm is ready to consider acquisition or a public offering, patents are the preferred method for signaling the firm's technological value. *Id.*

[145] *Id.*

[146] Mark Lemley, Doug Lichtman, & Bhaven Sampat, *What to Do About Bad Patents?,* http://www.law.uchicago.edu/files/lichtman/bad-patents.pdf (last visited May 20, 2008). Many internet patents award legal rights that far exceed the relevant value of the invention. *Id.* The USPTO receives large numbers of bad patents and is ill-equipped to deal with software patents. Evans & Layne-Farrar, *supra* note 42, ¶ 24.

[147] Mann, *supra* note 16, at 990.

[148] *Id.* Instances of cross-licensing are quite high among larger firms. *Id.* When two companies engage in the cross-use of each other's patent portfolio, a cold war scenario comes into being in the sense that "mutually assured destruction" will result from either firm initiating litigation. *Id.* at 991.

[149] Evans & Layne-Farrar, *supra* note 42, ¶ 54. Patents curtail whatever positive innovation effects they might have had by giving larger companies disproportionate power. *Id.*

1. Pooling

The patent pooling model instituted by OIN is a variation on the idea of cross-licensing.[150] Donated and purchased patents are centralized and made available for all open source developers.[151] This repository of patents helps the open source community combat the cross-licensing proprietary-software threat by establishing a basis for a level playing field.[152] A would-be plaintiff, who is himself guilty of infringement, may be hesitant to level accusations of infringement against smaller open source entities.[153]

Patent pooling also promotes economy by eliminating the need to divert resources to redeveloping preexisting ideas, which in turn boosts the quality of target software because it allows more focus to be placed on innovation.[154] The value of patent pools in boosting innovation and defending against infringement allegations will continue to build as membership grows and new patents are added to the existing pool.[155]

[150] *IBM Joins Open Invention Network, Promoting Linux*, IBM NEWS, Nov. 9, 2005, http://www.ibm.com/news/us/en/2005/11/2005_11_10.html. IBM, Red Hat, Novell, Philips, and Sony all announced the formation of, OIN, a new company dedicated to the continued growth of Linux. *Id.* The OIN network acquires patents and licenses them royalty free to others who agree not to assert their patent rights against Linux in exchange. *Id.*

[151] Open Invention Network, http://www.openinventionnetwork.com/about.php (last visited May 20, 2008).

[152] Stephen Shankland, *Open-Source Allies Go On Patent Offensive: Two Linux Allies Are Taking a Leaf Out of Their Opponent's Book as They Try to Prevent Software Patents From Putting a Crimp in Open Source*, CNET NEWS, Aug. 11, 2005, http://www.news.com/Open-source-allies-go-on-patent-offensive/2100-7344_3-5827844.html.

[153] *See* Haulbrook, *supra* note 2, para. 25 (stating that a company may abstain from bringing an infringement suit against someone in possession of a large patent portfolio for fear the would-be infringer will rail down destruction on its accuser).

[154] Patrice-Emmanuel Schmitz & Sébastien Castiaux, *Pooling Open Source Software: An IDA Feasibility Study* (2002), http://europa.eu.int/idabc/servlets/Doc?id=1977.

[155] *Id.*

2. Prior-Art Databases

The posting of open source projects on such popular websites as Sourceforge[156] and Freshmeat[157] will guarantee the source code will not be misappropriated, patented, and enforced against its own developers or downstream recipients.[158] Such assurance is secured because the material "published" in the repository immediately becomes prior art and thus represents a statutory bar to patenting,[159] leaving copyright law as the controlling legal authority.[160] Prior-art databases, used in conjunction with patent pooling, will aid in preserving the integrity of the open source software model.

a. Weaponizing the GNU GPL

The third incarnation of the FSF's GPL is armed with enhanced patent protection.[161] While only a direct challenge to the license will determine the effectiveness of the new measures, in theory they appear to constitute an effective upgrade from the previous version.[162]

With the explicit patent grant clause, a distributer of a GPLv3 licensed derivative work automatically grants a nonexclusive, royalty-free license for any patented contributions to all downstream recipients, shielding them from patent infringement claims.[163] The cross-licensing restriction clause will protect downstream users from third party cross-licensed patents.[164] If those patent licenses cannot be extended to downstream users, then the distributor loses his rights to the derivative work under the GPL.[165]

[156] Sourceforge, http://sourceforge.net/docs/about (last visited May 20, 2008). Sourceforge is an Internet repository that hosts over 100,000 ongoing open source projects that are freely accessible. *Id.* Source code is posted and updated as projects progress. *Id.*

[157] Freshmeat: About, http://freshmeat.net/about/ (last visited May 20, 2008). Fresh meat is another popular repository for open source projects. *Id.*

[158] Carver, *supra* note 16, at 463–64. To facilitate browsing and downloading, virtually every free software project posts its code on one of the various popular internet sites. *Id.* This method of operation is invariably the open source community's greatest protection from future patent threats. *Id.* This is because patents are only available if the novelty and nonobviousness requirements can be satisfied. *Id.*

[159] 35 U.S.C. § 102(b) (2006) (stating that a person cannot obtain a patent if "the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States.")

[160] *See* Russell Moy, *A Case Against Software Patents*, 17 SANTA CLARA COMPUTER & HIGH TECH. L.J. 67, 98 (2000) (stating that copyright protection is available where patent claims are precluded); *see also* Richard H. Stern, *The Bundle of Rights Suited to New Technology*, 47 U. PITT. L. REV. 1229, 1246 (1986) (stating that copyright protection attaches automatically upon creation of software).

[161] Sapna Kumar, *Enforcing the GNU GPL*, 2006 U. ILL. J.L. TECH. & POL'Y 1, 5 (2006); GNU Project: GNU General Public License Version 3 (2007), http://www.gnu.org/licenses/gpl-3.0.html.

[162] Kumar, *supra* note 161, at 4–5 nn.20 & 23.

[163] Zhu, *supra* note 75, paras. 19–21.

[164] *Id.* para. 22.

[165] *Id.* paras. 15, 19.

These incorporated safeguards have hardened GNU GPLv3 against patent threats. In conjunction with the other patent defenses implemented by the open source community, the open source software model should be able to stand its ground against the patent-wielding proprietary software industry bent on destroying it.[166]

III. Proposal

The policy behind the Patent Act is to provide incentive for innovation by rewarding public disclosure with time-limited monopoly rights for the inventor.[167] Because the legislation was drafted during the mechanical age, it has not adapted well to software despite its prior success with post-mechanical industries such as electronics and pharmaceuticals.[168] The problem has been exacerbated by the courts assuming a very liberal stand on the patent eligibility of software.[169] Balance may be restored by tuning copyright[170] and patent protection to maximize innovation and public disclosure and allowing open source and proprietary software models to coexist in harmony.[171] This can be accomplished by tightening the nonobviousness standard for building-block programs while allowing patentability for creative algorithms, requiring a more open prosecution process which includes disclosure of source code, shortening the patent term for software, and providing governmental support for keeping industry standards open.

A. Nonobviousness and Functional Claiming

Reassembling existing subroutines to perform new functions is the basis of software development.[172] This building block approach rarely applies to new mechanical devices that are usually not based on cumulative engineering and

[166] *See* Jason B. Wacha, *Taking the Case: Is the GPL Enforceable?*, 21 SANTA CLARA COMPUTER & HIGH TECH. L.J. 451, 477–78 (2005).

[167] Michael R. Taylor & Jerry Cayford, *American Patent Policy, Biotechnology, and African Agriculture: the Case for Policy Change*, 17 HARV. J. L. & TECH. 321, 339 (2004).

[168] Peter S. Menell, *A Method for Reforming the Patent System*, 13 MICH. TELECOM. & TECH. L. REV. 487, 495 (2007).

[169] Scott Elengold, *An Inquiry into Computer System Patents: Breaking Down the "Software Engineer"*, 61 N.Y.U. ANN. SURV. AM. L. 349, 351–56 (2005). The ease with which the USPTO grants software patents has resulted in many that are considered "absurd" by most software engineers. Allen Clark Zoracki, *When is an Algorithm Invented? The Need for a New Paradigm for Evaluating an Algorithm for Intellectual Property Protection*, 15 ALB. L.J. SCI. & TECH. 579, 586 (2005).

[170] Haulbrook, *supra* note 2, para. 28. The courts have been very reluctant to give copyright protection for computer software a broad interpretation. *Id.* para. 7.

[171] Yann Joly, *Open Source Approaches in Biotechnology: Utopia Revisited*, 59 ME. L. REV. 385, 405 (2007). The patent represents a carefully crafted agreement between the public and the inventor. SCHECHTER & THOMAS, *supra* note 110, at 9–14. The inventor trades his novel and useful invention for a limited time monopoly in an effort to promote research and development. *Id.* Dwindling competition and economic hardship are the result of shifting the balance of patent protection too strongly in favor of the patentee. *Id.*

[172] Zoracki, *supra* note 169, at 594.

designed from the ground up.[173] The distinction should be reflected in a 35 U.S.C. § 103 nonobviousness assessment. It is crucial that the USPTO employ qualified software engineers that understand software and how it is developed.[174] Software that results from creative genius should be patentable, but programs easily developed by ordinary software professionals tasked with a specific goal should not be patentable.[175] Phrased differently, any software solution that could have been easily coded by a skilled developer anticipating the corresponding problem should be considered obvious.[176]

In essence, software patents protect the ideas behind programs whereas copyrights protect the specific source code implementations of those ideas.[177] Particularly, for software, it is often the implementation that is innovative and crucial for proper functioning.[178] For example, reengineering algorithms for use in high-performance scientific and gaming applications to run faster and use less memory will allow for greater advancement despite hardware limitations.[179] With patents, development and implementation of such superior algorithms are barred because of functional claiming.[180] Therefore, programs that contain a novel new algorithmic technique should be patentable, whereas programs merely assembled from preexisting algorithms, short of creative genius, should not be patentable to allow for future improvement.[181]

The hiring of qualified software engineers alone is insufficient to eliminate bad software patents.[182] The USPTO must provide examiners with a searchable database that is both comprehensive and structured.[183] The body of prior art for software is vast and having such a complete database at their disposal, indexed by

[173] Evans & Layne-Farrar, *supra* note 42, ¶ 53.

[174] *See* Lance D. Reich, *One of Skill in the Art in Software Engineering: The Rising Tide*, 84 J. PAT. & TRADEMARK OFF. SOC'Y 269, 270–71 (2002).

[175] *See* Elengold, *supra* note 169, at 364–65; Zoracki, *supra* note 169, at 592.

[176] *See* Zoracki, *supra* note 169, at 594. Revealing an obvious way of assembling pre-existing algorithms to create new software is of no benefit to the public and therefore falls short of the purpose behind granting patents which is the dissemination of innovation to society. *Id.*

[177] *See* Mazur v. Stein, 347 U.S. 201, 217 (1954); *see also* Bradford L. Smith & Susan O. Mann, *Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents?*, 71 U. CHI. L. REV. 241, 256 (2004); Steven B. Toeniskoetter, *Protection of Software Intellectual Property in Europe: An Alternative SUI Generis Approach*, 10 INTELL. PROP. L. BULL. 65, 70–71 (2005).

[178] Richard S. Gruner, *Better Living Through Software: Promoting Information Processing Advances Through Patent Incentives*, 74 ST. JOHN'S L. REV. 977, 986–87 (2000).

[179] Zoracki, *supra* note 169, at 600–01.

[180] Moy, *supra* note 160, at 99. Although an innovative new algorithm may be superior in every way, it may still be barred because it performs the same function as a previously patented algorithm. *Id.* Where a method is unpatented, free competition among programmers inevitably results in improvement to that algorithm. Gruner, *supra* note 178, at 996.

[181] *See* Zoracki, *supra* note 169, at 605. As the power of computer processors quickly grows, improved algorithms will be needed to keep up. *Id.* The development of such algorithms will be effectively blocked for a period of up to twenty years if the function of the algorithm is covered by a patent. *Id.*

[182] *See* Susan Walmsley Graf, *Improving Patent Quality Through Identification of Relevant Prior Art: Approaches to Increase Information Flow to the Patent Office*, 11 LEWIS & CLARK L. REV. 495, 502 (2007) (stating that patent examiners lack sufficient time to perform adequate reviews of the prior art).

[183] *See* Smith & Mann, *supra* note 177, at 261.

algorithm type, will allow examiners to more efficiently evaluate the relevant prior art and screen applications for novelty.[184] An effort to provide such a database has been undertaken by IBM and should be fully supported.[185]

B. Improving Public Disclosure

Patent-eligible software should be subject to stricter scrutiny and disclosure requirements to better protect public interests. The process of prosecuting software patents should be open to third parties.[186] While ex parte prosecution has proven sufficient for non-software patents in the past, the USPTO is currently ill-equipped to deal with the growing number of software patent applications.[187] By allowing others to weigh in before patents issue, more bad patents will be screened out.

Patent examiners have a very limited amount of time to assess the patent eligibility of software.[188] Third parties with vested interests at stake have a strong incentive to commit considerable resources to scouring prior art and bringing relevant information to the attention of the examiners.[189] Considering that one-third of all litigated software patents are declared invalid,[190] allowing examiners an "extra pair of eyes" will increase efficiency and avoid costly reexaminations and litigation later on.[191]

When seeking a software patent, the applicant should be required to disclose the source code.[192] This will serve to both protect the applicant in an open examination process and ensure the public receives the full benefit of the innovative new software. Having source code on file would limit complications in a process that is no longer secret in the event that modification of the claims becomes necessary. Analogous to the written description requirement, source code disclosure would reveal exactly what subject matter the inventor was in possession of at the time of filing.[193] It would also ensure that the public receives more than just a series of flow charts in

[184] *See* Lindholm, *supra* note 138, at 96–106.

[185] *See* Seth H. Ostrow & Arianna Frankl, *Patent Quality Improvements in the Works at the USPTO*, 12 NO. 10 INTELL. PROP. STRATEGIST 3 (2006).

[186] *See* Shane Glynn, *Rationalizing Software Patents: Suggestions for a Livable System*, 28 HASTINGS COMM. & ENT. L.J. 287, 307 (2006) (stating that when it comes to challenging new patents, the market competition is always the most motivated player in finding disqualifying prior art for obviousness and novelty).

[187] Ian Ayres & Gideon Parchomovsky, *Tradable Patent Rights*, 60 STAN. L. REV. 863, 868–69 (2007).

[188] *Id.* On average, Patent examiners only have about eighteen hours to spend on each application. *Id.*

[189] Glynn, *supra* note 186, at 307.

[190] Webbink, *supra* note 1, ¶ 8.

[191] *See* Glynn, *supra* note 186, at 307 (proposing outsource of prior art searches to the software industry).

[192] Webbink, *supra* note 1, ¶ 28.

[193] *See* SCHECHTER & THOMAS, *supra* note 110, at 191. By requiring an enabling written description of the invention, the applicant reveals the extent of his invention as of the filing date which could be important if unforeseen future complications arise. *Id.*

exchange for patent rights.[194] Justification for mandatory source code disclosure would come from the best mode requirement under 35 U.S.C. § 112.[195]

Currently, pending patent applications need only be published eighteen months after filing if the applicant intends to file in a foreign country with a publishing requirement.[196] All pending software patent applications should be published irrespective of the applicant's foreign filing intentions.[197] This should occur after six months rather than eighteen, given the speed of software development. This shorter time frame will provide better notice of the intellectual property that is being claimed and help cut back on inadvertent infringement.[198] It also prevents unscrupulous applicants from exploiting the patent prosecution process to reap financial gain from submarine patents.

C. Special Software Patents

The fact that an entire generation of programs can cycle within a matter of months, in the continuous process of software development, supports the argument that the term for software patents should be shortened. Given the low overhead of software innovation and its limited "shelf life," a seven year term of protection would be adequate.[199] This will increase innovation by reducing patent congestion in two ways. First, it will reduce the number of filings, and second, the overlap with other patents more than seven years since issuance will be avoided.[200] Additionally, a

[194] *See* John T. Soma et al., *Software Patents: A U.S. and E.U. Comparison*, 8 U. BALT. INTELL. PROP. L.J. 1, 46 (2000).

[195] Section 112, ¶ 1 states, "The specification . . . shall set forth the best mode contemplated by the inventor of carrying out his invention." 35 U.S.C. § 112, ¶ 1 (2006); *see* SCHECHTER & THOMAS, *supra* note 110, at 196–200. The best mode is usually the way the inventor practices his invention. *Id.* The requirement compels the patentee to disclose to the public that which otherwise would be maintained as a trade secret. *Id.*

[196] 35 U.S.C. § 122; MUELLER, *supra* note 46, at 43.

[197] *See* Glynn, *supra* note 186, at 307. Currently, patent applications are kept secret if they either have not or will not be filled outside of the United States. *Id.* To remove the privacy provision for domestic applicants, Congress would have to pass legislation to modify 35 U.S.C. § 122. *Id.*

[198] John LaBarre & Xavier Gomez-Velasco, *Ready, Set, Mark Your Patented Software*, 12 RICH. J. L. & TECH. 3, ¶ 61 (2005). Because the likelihood of innocent infringement is growing exponentially with the large proliferation of software recently, many legal scholars are calling for software to be marked. *Id.* This presents a unique challenge though because programs and algorithms are not always distributed in packaging that will accept a permanent mark. James W. Soong, *Patent Damage Strategies and the Enterprise License: Constructive Notice, Actual Notice, No Notice*, 2005 DUKE L. & TECH. REV. 2, ¶ 4, (2005).

[199] *See* Zoracki, *supra* note 169, at 596.

[200] *See* Webbink, *supra* note 1, ¶ 4 (giving a hypothetical that if serious software patenting occurred since 1975, today's inventions would have been subject to patents for the better part of twenty years). Patenting software in earnest has only recently begun. *Id.* Since a full twenty years from has not yet come to pass, we have not yet crossed the threshold of maximum patent overlap. *Id.*

reduced term increases the odds that some utility will remain for the public when the monopoly rights expire.[201]

D. Open Source Concessions

Given that the open source development model calls for unrestricted access to software, the public would be better served if the proprietary software industry did not have monopoly control over industry standards.[202] The U.S. government should subsidize open source development of universal standards to ensure compatibility through uniformity.[203] The alternative is that each proprietary industry would seek to establish market dominance of its own standard in an effort to lock in consumers and retrieve royalties.[204] When consumers have to give up choice to maintain compatibility, the result is a fractured market.[205]

Although the patent defense clause has never been directly litigated, the court should allow the open source community to protect itself from the proprietary software industry's well-funded assailment. GPLv3 stipulates that anyone asserting patents against downstream users would lose his or her own license to that copyrighted work.[206] This clause, along with the patent-grant clause, should be upheld.[207]

[201] *See* Lindholm, *supra* note 138, at 92. The economic value of the innovation bestowed onto the public at the end of a patent term is typically lower for software than any other industry. *Id.* Software depreciates at a greater relative speed. *Id.*

[202] *See id.* at 127. Additionally, when payment is required for the use of patented standards, free software is unable to incorporate those standards to benefit society. *Id.*

[203] *Id.* The World Wide Web Consortium, responsible for developing Web protocols and guidelines, understands that free software betters society and approves only standards available without royalties. *Id.*

[204] *See* Mann—*Commercializing*, *supra* note 138, at 8–9 (stating that the interests of a firm owning a standard would be in conflict with the interests of competing firms attempting to provide products and services that conform to that standard).

[205] Zoracki, *supra* note 169, at 586. The existence of too many questionable and overly broad patents would leave a digital world carved up into little pieces where there is little power to link people, communities and ideas. *Id.*

[206] FSF: Frequently Asked Questions About the GNU GPL, http://www.fsf.org/licensing/licenses/gpl-faq.html (last visited May 20, 2008). If a company, for example, modified GPLv3 copyrighted software with its own code and implemented the resulting software on its own servers as well as distributing it to others, any later assertion of patent rights against downstream users by the company would also invalidate the software running on the companies servers. Mann—*Commercializing*, *supra* note 138, at 20.

[207] *See* Haulbrook, *supra* note 2, para. 7. Copyright protection for software does present some challenges. *Id.* In contrast to patents, there is no one governing federal circuit with nationwide jurisdiction over copyright cases. *Id.*

CONCLUSION

The courts have set forth a very liberal interpretation of patent eligibility for software, resulting in a thicket of over 100,000 issued patents in effect today.[208] As of 2005, software patents claimed over fifteen percent of the total patent share.[209] Compounding the problem, software companies have responded defensively by securing even more patents[210] in an effort to achieve more favorable bargaining positions.[211] This threatens the open source model, which is not predicated on the pursuit of a lucrative monopoly, and stifles innovation.[212]

Several logical changes hold promise in restoring equilibrium to a troubled system.[213] The requirement of novelty and nonobviousness must be strictly applied by defining the terms from the vantage point of the skilled software engineer. This tighter standard must actually be implemented by staffing the USPTO with such qualified engineers and searchable databases. The patent prosecution process should also be open to third parties who may have the additional vested interest and resources to catch potentially bad patents before they issue, avoiding costly challenges. Furthermore, because improving the speed and memory requirements for existing algorithms goes to the heart of computational evolution, such innovation should not be blocked by the restrictions of functional claiming.

Software patent applications should include the full disclosure of source code to meet the best mode requirement. This will protect the applicant in an open examination process by showing he was actually in possession of the claimed material as of the filing date and also secure the full value of the innovation for the public. Because software has such a short generational cycle, patent terms should be reduced to seven years and publication should occur for all software patent applications six months after the filing date. This will guard against submarine patents, inadvertent infringement, and maximize innovation.

The suggestions outlined above will substantially reduce the number of software patent applications, which in turn will allow for more thorough examinations by

[208] Smith & Mann, *supra* note 177, at 263; Zoracki, *supra* note 169, at 585; Webbink, *supra* note 1, ¶ 5.

[209] Zoracki, *supra* note 169, at 585.

[210] Webbink, *supra* note 1, ¶¶ 5–6. Microsoft currently files several thousand patents per year. *Id.* IBM is granted about a thousand patents per year. Haulbrook, *supra* note 2, para. 14.

[211] Barton E. Showalter & Jeffery D. Baxter, *Strategic Use of Software Patents*, *in* FEBRUARY-MARCH 1999: 19TH ANNUAL INSTITUTE ON COMPUTER LAW 1999, at 1057, 1076–77 (PLI Pats., Copyrights, Trademarks, & Literary Prop., Course Handbook Ser. No. 885, 1999), *available at* WL, 547 PLI/Pat 1057. Among large sophisticated software firms, for every three patents that are licensed for revenues, seventeen are used defensively to maintain freedom of action. Mann—*Commercializing*, *supra* note 138, at 22. Compare this with the pharmaceutical industry where only one patent is used defensively for every three that are licensed for revenues. *Id.* Those patents that are used defensively do not support innovation. Menell, *supra* note 168, at 506. Such games of mutually assured destruction, where each company leverages their patents against the other, have been characterized as a misuse of the patent system. Haulbrook, *supra* note 2, para. 25.

[212] Webbink, *supra* note 1, ¶¶ 7, 9–10. Studies seem to indicate that firms are treating patents like investments, substituting them for research and development. *Id.* With a large enough patent portfolio, chances are good that someone else's invention may be taxed for including protected content. *Id.*

[213] See *id.* ¶¶ 26–28.

reducing the USPTO's workload.[214] Shrinking the patent thicket will also allow smaller software companies and open source developers to survive.[215]

Open source software is uniquely suited for the establishment of industry standards.[216] Supporting open source standards will ensure uniformity and compatibility in the marketplace because it is distributed freely without restriction or expectation of royalties.[217] Healthy competition between open source and proprietary software stimulates innovation and provides greater consumer choice. Because this is the very policy behind the Patent Act, courts should give teeth to the patent-grant and patent-defense clauses in the licenses under which open source software operates.

[214] *See* Brian E. Mack, *PTO Rulemaking in the Twenty-First Century: Defining the Line Between Strategic Planning and Abuse of Authority*, 75 FORDHAM L. REV. 2105, 2106–07 (2007). Reduction in the examiners workload will also reduce pendency and the problems that surround it, such as reduced terms and submarine patents given a 35 U.S.C. § 122(b)(B)(i) publication exception. *Id.* Pendency is the time between the filing of the application and the issuance of the patent. *Id.* The pendency on software patents averaged to over three years for 2005. U.S. PATENT AND TRADEMARK OFFICE, PERFORMANCE AND ACCOUNTABILITY REPORT FISCAL YEAR 2005 tbl.4 (2005), *available at* http://www.uspto.gov/web/offices/com/annual/2005/060404_table4.html. This is over a year longer than the average pendency for non-software patents. *Id.*; Zoracki, *supra* note 169, at 587. 35 U.S.C. § 154(b) guarantees at least a seventeen year patent term from date of issue. MUELLER, *supra* note 46, at 18–19. If, due to the fault of the USPTO, the pendency period is greater than three years, time will be added to the term accordingly. *Id.*

[215] *See* Webbink, *supra* note 1, ¶ 14. Smaller start up software companies are most at risk from the patent thicket. *Id.* When faced with potential litigation, they have neither the patent portfolio to bargain their way out nor the capital to fight their way through. Menell, *supra* note 168, at 507.

[216] *See* Mann—*Commercializing*, *supra* note 138, at 8–9 (stating that a firms would be hesitant to support through goods and services a third party's patented for profit standard over their own).

[217] *See* Bennett M. Sigmond, *Free/Open Source Software Licensing—Too Big to Ignore*, COLO. LAW., Dec. 2005, at 89, 90 (giving various advantages of open source software, such as being free of fees and preventing vendor-lock-in).