Open Source Software and Software Patents

Finding the Common Ground in a Patent Pool

Jason V. Morgan
Candidate for Bachelor's degree in Computer Science
University of Utah School of Computing
Salt Lake City, UT
morgan@cs.utah.edu

ABSTRACT

Software patents arose from a history of turmoil with established legitimacy in the business and legal worlds. Now they threaten the future of the Open Source Software (OSS) community, especially with the increased importance of software standards and the change in attitude in the organizations forming these standards. The OSS community does not exercise much influence over the policymakers who have legitimized software patents because of differences between the two groups' motivations; the OSS community is not speaking the language of policymakers. There are many possible actions the OSS community could take, but an OSS patent pool is the best way for the OSS community to leverage both the patent system and the OSS culture to help protect the future of OSS.

1. BACKGROUND

Software patents have a long history, coming out of obscurity only during the past two decades. Early in that history, the position of United States policymakers was unclear regarding, and in some cases even hostile towards, software patents. But since then, the attitude of these policymakers has solidified in favor of software patents. Today software patents have credibility, not only in the world of law, but also in the world of business.

1.1 Ancient software patent history

The debate about the patentability of software is not new. Software patentability has been an issue for at least the 50 years since Congress passed The Patent Act of 1952, an act codifying the scope of patentable subject matter in the United States Code (U.S.C.) as "Section 101":

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title[2].

Many people believed that software fell outside of the scope of inventions patentable under Section 101. The 1966 Presidential Commission on the Patent System gave weight to this belief with its recommendation that patents should not be issued for computer programs[1].

The Supreme Court struck down the first software patent brought before it in *Gottschalk v. Benson*[23], inadvertently sending many the message that software was unpatentable[66, p. 134]. The Supreme Court's decision stated that "it is said that the decision precludes a patent for any program servicing a computer...we do not so hold[23, p. 71]." This left open the possibility that software patents could be considered valid under common law, but the credibility of software patents was still damaged.

A software patent "dark age" followed. Few software patents were issued during the 1970s and 1980s because "many attorneys believed software could not be patented and counseled their clients accordingly[66, p. 209]." Many of the software patents issued were disguised as hardware patents in order to push them through a hostile United States Patent and Trademark Office (USPTO)[27].

The USPTO would not even hire patent examiners with degrees in computer science until 1994. This was partially due to the bias towards hardware people, leftover from the earlier days of the computer industry when hardware products were valued more than software products[66, pp. 393–394]. It was also a byproduct of the USPTO's reluctance to accept software as patentable subject matter. This approach helped to solidify an anti-software patent attitude among software developers, particularly because the USPTO was ill equipped to determine if the requirements of "novelty and nonobviousness" were met for many issued software patents.

A lack of credibility was not the only thing working against software patents early in the software industry's history. A patent is the grant of a legal monopoly by the government. In the 1930s, Government watchdogs became zealous about using antitrust law to reduce the anticompetitive effects of patents[57, pp. 37–38]. In the 1970s, the Department of Justice's (DOJ) introduced the "Nine No-Nos," patent licensing practices that the DOJ considered anticompetitive. This focus on antitrust law greatly reduced the extent to which patent monopoly power could be exploited by the assignee[18, pp. 284–285], thus reducing the incentive for companies to stand behind software patents.

The lack of a Supreme Court decision clearly favoring the patentability of software, the USPTO's software patent hostility, and the DOJ's antitrust zeal all created an atmosphere in which software patents would have little value. This era did not last long.

1.2 "Anything under the sun..."

A dramatic shift in judicial opinion took place during the 1980s that suddenly gave software patents credibility:

> The uncertainties injected into patent law by [the Supreme Court decision in *Gottschalk v. Benson*] have been largely erased by the Supreme Court in the landmark decision in *Diamond v. Chakrabarty*[11], which held that Congress, in drafting the Patent Act of 1952, intended statutorily patentable subject matter to include "anything under the sun that is made by man[11, p. 309]." Having crystallized the issue in *Chakrabarty*, finding that anything created by humanity is potentially patentable, the Supreme Court thereafter ruled in *Diamond v. Diehr*[12] that software is patentable subject matter[66, p. 30].

Policymakers outside of the Supreme Court began to shift their positions too during the 1980s. "The Antitrust Division began to question the theory underlying the Nine No-Nos, focusing on the principle that unconstrained patent licensing increases the value of patents and encourages licensing and innovation[18, p. 286]."

Judicial support for software patents, along with a reduced risk of facing an antitrust suit for leveraging patent monopoly power, made software patents much more valuable. This led to a dramatic increase in the number of software patents issued each year.

By 1995, even the USPTO had fully accepted software patents and was "preparing guidelines for the patent examining corps, advising the corps that computer programs embodied in a tangible medium, such as a floppy diskette, are patentable subject matter under 35 U.S.C. 101 and must be examined under 35 U.S.C. 102 and 103[38, p. 2]." The final examination guidelines for computer-related inventions show that, at least within the USPTO, software is considered patentable:

> The utility of an invention must be within the "technological" arts. A computer-related invention is within the technological arts. A practical application of a computer-related invention is statutory subject matter. This requirement can be discerned from the variously phrased prohibitions against the patenting of abstract ideas, laws of nature or natural phenomena. An invention that has a practical application in the technological arts satisfies the utility requirement[15, pp. 2–3].

With the USPTO's acceptance of software patents, the debate among U.S. policymakers about the patentability of software was over. Section 101 was no longer the main issue facing software patents.

By this time, leading companies had already embraced the use of software patents, if not always for exploitation. Software patents were acquired so that they could be used against other software patent holders during patent infringement cases. This "Mutually Assured Destruction" form of defense became common during the period of the Nine No-Nos[57, pp. 38–46].

Even Oracle had filed four software patent applications by 1994, when Senior Vice-President Jerry Baker testified in the USPTO's public hearings on software patents that "at Oracle we believe that patents are inappropriate means for protecting software and [we] are concerned that the patent system is on the brink of having a devastating impact on the software industry[69, p. 25]." Today, Oracle has 327 issued patents.

Today, software is patentable in the United States, a situation accepted by both the legal world and by the business world. Many believe that software should not be patentable. But in the decision-making community, software patents have credibility.

2. THE THREAT TO OPEN SOURCE SOFTWARE

With software patents on the rise, the Open Source Software (OSS) community faces great challenges in the years ahead because of software patent. "Patents are fully capable of crippling Open Source development to the point where it is no longer competitive with advances in proprietary software[57, p. 193]."

2.1 Standards and the handcuffs of compatibility

The software industry is awash with standards, both formal standards set by standardization groups and de facto standards set by a single entity with market power. OSS must implement many of these standards to be compatible with existing proprietary software, to be useful for many users. Software patents could jeopardize the OSS community's ability to create software that is compatible with existing proprietary software.

2.1.1 LZW: A glimpse of the future

Perhaps the most frequently cited example of a software patent that has stopped the OSS community from fully implementing a standard is the LZW patent[4]. "LZW is a variant of the LZ78 process that was described in a paper by Terry Welsh of Sperry (now Unisys) in 1984. CompuServe adopted it for use in the GIF format shortly afterwards[40, p. 179]." Because the LZW compression algorithm is part of the GIF standard, any implementer of GIF-writing software must face the possibility that the implementation infringes on the LZW patent claims.

Some developers found ways to create GIF data that could be read by an LZW decompresser, but that did not use the patented LZW algorithm. GD, a popular graphics library, once used one of these algorithms to allow users to create GIFs[43]. However, the author dropped this feature in version 1.6 to "completely avoid the legal controversy regarding the LZW compression algorithm used in GIF[s][7]."

A big part of this decision probably can be attributed to the aggressiveness of Unisys in its assertion of LZW patent rights, an aggressiveness that had been growing for years[5]. This aggressiveness eventually led to the Unisys push to obtain licensing fees from users who create GIFs using software that does not license the LZW patent[41, 72].

The LZW patent has inspired a tremendous response from developers. Both the GNU's Not Unix (GNU) project and

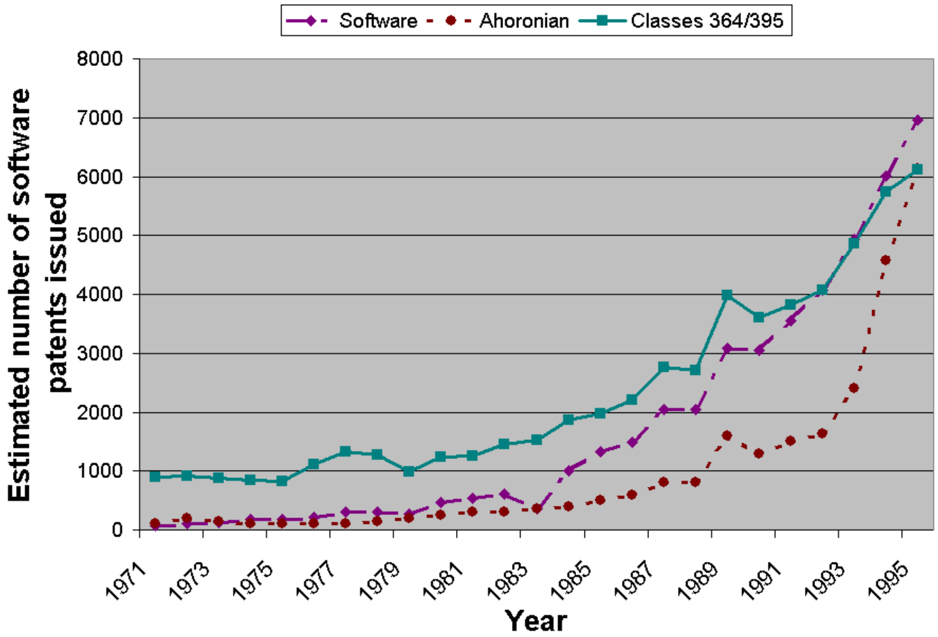

Figure 1: Results from three different methods of estimating the number of software patents issued per year between 1971 and 1995. Based on data in the "1995 Software Patent Statics[3]."

the Free Software Foundation (FSF) provide links next to almost every image on each web site explaining that the sites do not use GIFs because of the LZW patent[68, 76]. Thousands of email "flames" have been sent to Unisys protesting the LZW patent. There even exists a web site advocating the eradication of GIF use on the Internet[8].

What makes the LZW patent an issue in the OSS community has little to do with the algorithm itself—after all, better compression algorithms have since been developed, many of them without any patent burdens. The LZW patent is an issue because a developer must license the LZW patent in order to write software that creates GIFs without being threatened by Unisys. The GIF standard, paired with the refusal of OSS developers to license the LZW patent on Unisys's terms, is what makes this patent a problem for OSS.

2.1.2 De facto standards and the future

With the increasing need for software products to communicate with each other, the need for common languages, for standards, is becoming increasingly vital. However, just like with the GIF standard, it is possible for a patent holder to have staked out a piece of the intellectual property landscape covering an important standard.

Often, the market power behind a software product makes the formats, protocols, and Application Programming Interfaces (API) of the software product a de facto standard. Unfortunately for the OSS community, it is easy for the original developer of a product to obtain very narrow patents that cover changes in the formats, protocols, and APIs of the product. If the changes are widely distributed, then other software developers must infringe on these patents in order to create compatible software. This can give the original developer almost absolute control over competing products.

For example, an MS-DOS emulator that implemented a fully compatible API for accessing long file names would likely infringe on one of Microsoft's patents. One patent is so narrow that many of its claims specify the hexadecimal numbers used to select new functions[39].

The Virtual File Allocation Table (VFAT) file system implementation in Linux could already be infringing on another Microsoft patent that deals with the problem of implementing a long filename system on top of a short filename system, a problem Microsoft created in the first place. This patent makes four claims, including:

1. In a computer system having a processor running an operating system, a method comprising the computer-implemented steps of:

(a) storing in the memory means a first directory entry for a file wherein the first directory entry holds a short filename for the file, said short filename including at most maximum number of characters that is permissible by the operating system;

(b) storing in the memory means a second directory entry for a the file wherein the second directory entry holds a long filename for the file and wherein the second directory entry includes an attributes field which may be set to make the second directory entry invisible to the operating system and the step of storing the second directory entry further comprises the step of setting the attributes field so that the second directory entry is invisible to the operating system, said long filename including more than the maximum number of characters that is permissible by the operating system; and

(c) accessing the first directory entry with the operating system[55].

It would be very easy to create a non-infringing method for solving the problem of allowing long filenames on a short filename file system. However, no other solution would be useful because any other approach would be incompatible with the de facto standard set by Microsoft. De facto standards like VFAT provide aggressive companies many opportunities for leveraging software patents against competing OSS.

2.1.3 *"The effect of patents and copyright in combating Linux remains to be investigated." —Microsoft Halloween Documents*[71]

During the 2001 O'Reilly Open Source Convention, Craig Mundie, a Senior Vice President at Microsoft, participated in a "Shared Source v. Open Source" panel. During the discussion, Carl Holden asked Mundie:

> . . .I think there's little debate in this room, probably even from you and others with Microsoft, that a lot of software patents are pretty ridiculous. But Microsoft, I'm sure, holds a lot of them, and you expressed a willingness to have Microsoft enforce them, even when the violator is an open source programmer. Do you agree with that?

Mundie's answer:

> Absolutely[64].

It is not surprising that Microsoft provides an example in which software patents are being used specifically to target the OSS community. Recently, Microsoft released a technical reference to its Common Internet File System (CIFS) access protocol, which is part of the Server Message Block (SMB) protocol used by Windows based networks.

The royalty-free license for the CIFS technology reference grants the licensee rights to use use software patents needed to implement CIFS[59]. However, it prohibits sublicensing using the GNU General Public License (GPL)[19] or the Lesser GNU General Public License (LGPL)[20], two popular licenses for OSS.

> This caveat is not entirely unexpected as it enables Microsoft to ensure that its CIFS patents cannot be converted to the GPL, and protects the company's intellectual property. However, by specifically targeting the GPL, Microsoft could effectively kill off the open-source Samba project's implementation of SMB[9].

Of course, forcing Samba, "the free SMB and CIFS client and server for Unix and other operating systems[75]," to become incompatible with Windows would not cause any Microsoft executives to shed tears. Much of the popularity of Linux is due to Samba, which "essentially allows people to remove NT servers[58]."

What makes this license even worse for the OSS community is that the legal community could consider it Reasonable and Non-Discriminatory (RAND) because anyone can license the CIFS patents at little cost. The license does not even discriminate against many OSS licenses since many of these licenses do not require that "in any instance that other software distributed with software subject to such license (a) be disclosed and distributed in source code form; (b) be licensed for purposes of making derivative works; or (c) be redistributable at no charge[59, §1.4]."

Microsoft is not the only entity with the power to leverage its software patents against OSS. A handful of the leading computer industry companies have already been granted thousands of patents. Many smaller entities also have software patents. It is highly likely that some of the claims in these patents are infringed upon by some OSS. Microsoft, which has a history of extremely aggressive business tactics, is just the most likely to use its patented technology to close markets to OSS.

Company	Total	Company	Total
IBM	18327	Micron Technology	1981
Motorola	10843	Sun Microsystems	1938
Texas Instruments	5470	National Semiconductor	1699
Xerox	4919	Microsoft	1410
Hewlett-Packard	4287	NCR	1354
Intel	3614	Tektronix	1279
Lucent Technologies	3487	Compaq	1245
Unisys	2702	Apple Computer	1145
AMD	2624	LSI Logic	1009

Table 1: Leading North American computer companies with more than 1000 patents granted under the electrical classes between January 1, 1977 and December 31, 2000 and the total number of those patents. Based on "Top 100 Computer Industry Companies[46]" and the USPTO's "Technology Assessment and Forecast Report: Electrical Classes[67]."

2.1.4 *Formal standards bodies & the compromise being made*

The semiconductor industry recognizes that "decisions by a dominant firm to alter its technology often lead to a de facto standard for an entire industry" and that "change in and of itself is not anti-competitive although change can be effectuated for anti-competitive purposes[24, p. 7]." The software industry is no different in this respect.

Fortunately, formal standardization bodies, such as the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO), create a mechanism for creating standards that cannot change because of the whims of a dominant firm. These standards give independent developers a reference for creating software that is compatible, while also giving users a resource for selecting software that best fits the needs of the user community. As product compatibility with competitors becomes more important in the software industry, the importance of these standards groups will also grow[16, p. 249].

Formal standards can still fall prey to the "embrace, extend, and extinguish" strategy—a strategy for which Microsoft is notorious[45]—when a dominant firm is involved. Standardization bodies also lose their ability to facilitate coordination if all of the parties involved would get more value from going with their own systems rather than coordinating[16, p. 240]. Nevertheless, formal standards still have value as a rallying point for users with demands for software compatibility.

A relatively recent trend in software industry standardization organizations is a move away from requiring participants to give up the right to demand royalties for the use of their respective patents. Instead, these organizations are beginning to simply require disclosure of patents relevant to a standard and they have begun experimenting with requiring participants to license relevant patents on a RAND basis.

One high-profile example of this is the W3C's proposal for a patent policy that would require all W3C members to "make a legally-binding commitment to license patent claims essential for implementing a W3C Recommendation on RAND terms[73]."

The policy meant that, even if a W3C member was planning on enforcing a software patent to "collect rent" for the use of a particular technology, the W3C could still issue a recommendation using that technology. Public outcry to the recommendations was incredible. Developers wanted the W3C recommendations to be Royalty-Free (RF). A large number of responses claimed that "RAND will hurt open source developers," that "RAND discriminates against the poor," and that the goals of the draft policy would go "against the spirit of the Web/Internet[51]." However, the W3C would not have been the first standards organization to issue standards with RAND licensing requirements. One person even commented in a response to the draft policy:

> If a patent is essential to the creation of a standard, other organizations exist that can be used to propagate it; the ISO and ANSI for instance, ECMA, even the IETF (though they have been rather cautious in using their RAND option for software). That such a standard will *not* be endorsed by the W3C will hopefully constitute a drawback to such options and encourage the RF licensing of patents and competition on product quality rather than size of legal staff[13]."

The eventual response from the W3C to the flood of public comments was the release of an updated version: the "Patent Policy Working Group Royalty-Free Patent Policy[74]." A cursory examination of the new draft policy gives one the impression that the W3C has reversed its position on allowing its members to hold intellectual property rights over recommended standards.

Indeed, the new RF policy states that the "W3C will not approve a Recommendation if it is aware that Essential Claims exist which are not available on Royalty-Free terms[74, §1.]." But, the drafters of this policy have found a way to create a loophole that restricts the W3C members less than the previous draft policy, yet does not raise a vocal response from the OSS community.

The older draft policy allowed members to opt-out of RAND licensing of "essential claims," patent claims that would be infringed by required portions of a W3C recommendation, but this had to be done within 60 days of the publication of the "last call working draft[73, §8.1]" and covered all W3C members. Under the new draft policy, a member can opt-out of RF licensing obligations either through disclosure or simply by not participating in the Working Group drafting a recommendation[74, §2.2].

This is not an improvement. The 2001 draft policy restricted all patent holders in the W3C from unreasonably exploiting their intellectual property. The 2002 draft policy only restricts patent holders in a particular working group; it does not restrict all patent holders in the W3C. It restricts the working groups themselves because a recommendation cannot be made that requires the licensing of patents, even if RAND.

Working groups could end up making more parts of their recommendations optional in order to make patent claims unessential. This would reduce the value of recommendations, possibly resulting in incompatibility among different implementations, even those that conform to a recommendation. Yet, W3C members would still be able to hold intellectual property rights over the optional sections of recommendations, thus allowing them to shut out groups such as the OSS community.

These rights could be very important if an optional feature is widely used. For example, a browser does not have to be able to read GIFs to be a fully compliant HyperText Markup Language (HTML) reader[29]. Nevertheless, a graphical browser that did not implement GIF decompression (which is fortunately not patented) would be considered incomplete by users because GIFs are still widely used.

Unlike the previous draft policy, this new policy could discourage W3C members from participating in Working Groups because "a company might profit from refusing to participate in the standard-setting process, in the hope that the resulting standard will nonetheless (perhaps inadvertently) infringe on the company's patent[63, p. 22]." This could reduce the technological expertise readily available to various Working Groups, resulting in less useful standards.

The W3C has an interest in seeing as much Internet technology as possible available on a RF basis, but the W3C has shown that it is willing to compromise. Many standards bodies are willing to make concessions to the intellectual property rights of their members and are likely to make even greater concessions in the future.

2.1.5 But why compromise?

Standards bodies have a vested interest in getting their members to license useful technology on a RAND basis. Part of this interest is maintaining the competitiveness of the released standards themselves in the face of a more patent-savvy generation of firms.

Another part of this interest is fulfilling the promise standards bodies make to advance and disseminate technology. It has been noted that in standards bodies for communication technologies "the earlier approach of public ownership and control of technical communications systems created significant limitations to the introduction of new technology and is now in decline[34]."

An uncompromising RF stance by standards bodies takes away the incentive for patent holders to come to the table. The intellectual property rights held are no less valid when the patent holder is not involved in standards setting, but awareness of relevant claims is limited when patent holders are not involved in the standardization process.

Pushing away patent holders, who often are influential industry leaders, also reduces the credibility of important standards because they are less likely to be adopted by those leaders. When important standards lack credibility and are not adopted, developers and consumers lose out on the benefits of standardization. Markets are then left to be dominated by de facto standards, which are more vulnerable to patent control than formal standards.

Standards bodies also face the need to balance antitrust considerations when deciding how to approach licensing requirements for members' patents. An organization requiring its members to license certain patents to others would likely be treated the same as a patent pool under antitrust law. It is likely that the Department of Justice still holds that:

> The most obvious threat to the innovation incentive is if pool members have to give royalty-free licenses. But even if the pool agreement provides for a "reasonable" royalty, the actual royalty for new inventions may be artificially low if it [is] set by the rest of [the] pool—i.e., by the innovator's competitors[33, p. 12].

These bodies have to deal with the threat of antitrust action if members are forced to license their patents, even on a RAND basis:

> It is well noted that many standard-settings organizations are wary of sanctioning any specific agreement regarding the magnitude of licensing terms for fear of antitrust liability, as such agreements might be construed as "price fixing." Perversely, by leaving the precise licensing terms vague, this caution can in fact lead to *ex post* hold-up by particular rights holders, contrary both to the goal of enabling innovation and to consumers interests[63, p. 10].

Potential anticompetitive effects can stem from non-RF licensing, if the standard of reasonableness is particularly costly[32, p. 36]. However, most of the software industry would likely benefit from a more flexible approach to dealing with patent licensing issues in standards bodies, especially if these bodies adopt and enforce a truly reasonable standard for RAND licensing.

2.2 Falling behind

RAND licensing often is not considered reasonable and non-discriminatory in the eyes of the OSS community. The definition of an "Open Source" license states right away that a licensee shall have the right to distribute the product without having to pay royalties[48, §1]. However, it is virtually impossible to meet this restriction if a patent holder requires the patent licensee to pay a royalty for each copy of the software distributed.

With more software patents being issued every year, OSS products face a greater risk of infringing on the intellectual property rights of an aggressive patent holder. The only way to completely avoid this risk is for OSS developers to stick to implementing software using 20-year-old innovations.

The likelihood of RAND licensing becoming necessary to fully implement important future standards means that OSS will be unable to meet the needs of many users. Some forms of RAND licensing, such as requiring patent grant backs to the licensor for patented improvements, would have little effect on OSS. However, under the current definition of "Open Source," even a royalty of a penny per distribution would literally shut out OSS developers from using a patented innovation.

Being forced to stay behind the cutting edge of technology would seriously damage the OSS community. Commercial investment in OSS would decrease, demoralizing the OSS community worse than the recent stock crash that wiped out many RedHat paper fortunes. Users would find OSS less able to meet their needs as changes in technology take place.

The importance of OSS being able to meet user needs should not be underestimated. The OSS community is fed by a gift culture in which a "good reputation among one's peers is a primary reward[53,p. 14]."

A decreased user pool would also mean that fewer developers would want to develop OSS, leading to a further decline in the ability of OSS to meet users' needs. The OSS community would not die, but it would fall far short of its full potential.

2.3 A unique risk

OSS developers cannot just ignore software patents, especially since there are some unique aspects of the OSS community that make it particularly vulnerable to infringement action.

2.3.1 The downside of being open

By definition, a licensee of OSS has access to the source code. This is OSS's greatest strength, but it is worth pointing out that—unlike with proprietary software in which the source code is a trade secret—patent holders do not have to go through discovery to verify whether a patent is being infringed by OSS or not. Even a study contracted by the European Commission to study software patents noted "in the US, it is safer for a company to keep source code secret rather than disclosing it in order to prevent software patent disputes[54, p. 18]."

2.3.2 Being SLAPPed around

Given the legal-wary nature of the OSS community, it is possible that a patent holder would not even bother trying to make a case against a developer. The threat of an infringement suit would be enough to push OSS developers out of a market, even if the infringement case were just a bluff.

Consider the LZW patent, the most notorious software patent and the most damaging. Unisys has only litigated this patent once, against Corel, settling out of court. OSS developers have stopped distributing GIF writing code, often code that did not use the LZW patent, merely because of the threat of being sued.

One does not even have to look at software patents to see OSS developers throwing away hundreds of hours of work because of an aggressive intellectual property holder. The Tetris Company, which claims to own the copyright to the "look-and-feel" of Tetris, managed to convince several independent developers of Tetris-like games to stop distributing their software. The company did not even have to file any lawsuits or obtain any injunctions to convince developers to stop competing[10].

It is understandable that a skilled OSS developer, for whom writing code may be a pleasure, would balk at incurring the expense of any legal defense. Instead of throwing support behind organizations such as the Electronic Frontier Foundation, OSS developers simply say that they do not have the resources that the big companies have—that they cannot afford to fight.

But OSS developers have to learn to defend themselves, because software patents are not going away.

3. WHY SOFTWARE PATENTS WILL NOT GO AWAY

Policymakers are going to continue to support software patents indefinitely. This community is no longer even seriously debating the patentability of software based on Section 101. Today, business method patents have taken center stage in the patent debate.

Even with business method patents, one can see that policymakers are in support of strong intellectual property protection. Representative Howard L. Berman stated, "I am not taking a final position as to whether business methods should be patentable. . .I tend to think they should be, but I could be persuaded otherwise[6]." This statement was part of his introduction for a bill that, among other things, would have expanded the definition of obviousness used for determining whether or not a patent should be granted to limit the granting of "e-commerce" patents[28].

3.1 Innovation as king

United States policymakers want to see rapid innovation, which is "the defining characteristic of the information economy[62]," and they are willing to sacrifice the OSS community if they believe that doing so will lead to increased innovative output. The problem is that the perception held by policymakers is that OSS does not produce much innovative output; that OSS mostly is about making non-proprietary copies of commercial systems. From a policymaker's perspective, this means that OSS can harm innovative output by reducing the value of innovations.

Unfortunately, many of the most popular OSS projects do little to refute this perception. For example, the GNU Manifesto states that "GNU. . .is the name for the complete Unix-compatible software system which I am writing so that I can give it away free to everyone who can use it[21]." This paints a portrait of "copycats," altruistic ones with a sense of community, but nonetheless, copycats instead of innovators.

More evidence that helps perpetuate this belief comes from the concern raised over the wasted energy that the OSS community puts into projects trying to compete with Microsoft on the desktop instead of making a better development environment[42]. While one can identify many incremental improvements made by the OSS community, there is little evidence that the OSS community is producing significant innovations.

Certainly, if there is much evidence, it has not been reaching policymakers. The 2001 Economic Report of the President only cites OSS once, not as a source of innovation or as an economic stimulus, but as a byproduct of advances in technology:

> Developments in information technology, meanwhile, have made possible entirely new R&D processes that further challenge the traditional centralized models. "Open-source" software design, which encourages users to modify the source code of a program and to share these improvements with others, has become increasingly widespread. Tens of thousands of programmers in the United States and abroad have contributed to open-source programs for such widely used products as Internet server software, e-mail routing software, and even some personal computer operating systems. Widespread Internet access has led to a dramatic acceleration in open-source activity, despite the fact that open-source programmers typically do this work without pay and distribute their source code for free. They may be motivated by reputation, which can lead to better future job offers and greater respect among peers, or by the sheer pleasure of solving the problem.[14, p. 114]

Policymakers see OSS as a development model, one that does not necessarily create the incentives needed to drive innovation. Since OSS has the potential to reduce an intellectual property's value, policymakers are unlikely to actively protect OSS from the interests of commercial software companies.

3.2 The losing battle

The OSS community could continue doing what it has done since at least December 1990, when members of the League for Programming Freedom formally adopted a stance against software patents[26]: complain. However, as long as OSS and innovation are not synonymous, the OSS community efforts will be futile.

It is hard to claim that the OSS community is not a vocal one. Advocacy/news sites such as slashdot.org and Kuro5hin.org give thousands of developers and users a potential launching platform for the latest OSS crusade.

The power of this OSS force can be seen in the tremendous response that Netscape received when it called for help in finding prior-art citations in a lawsuit brought against Netscape by Wang. Another example occurred a little over a year later, this time with the W3C calling for help to defeat a patent held by Intermind[57, pp. 186–187].

While the OSS community has shown great mobilization capabilities, the efforts of the community have been ineffective.

3.2.1 Economic power

Members of the OSS community have been active participants in many important hearings. However, their voice tends to be given less weight than the commercial entities that have also participated in these hearings.

For example, in the 1994 Software Patent Hearings before the USPTO, many individuals, usually representing either themselves or small software companies, testified quite vocally against software patents. But this was countered by the individuals representing large corporations and small start-ups who testified on behalf of software patents[66, 31], giving the argument in favor of software patents the weight of the supporters' "economic power."

The vocality of the OSS community was even more apparent in the study European Commission's study on the patentability of software. The study noted that 91% of the responses solicited opposed software patents, with a large proportion representing the OSS community. However, this same study also noted that an argument could be made, based on the economic power of the organizations supporting patents, that the pro-software patent group had an "economic" majority[54, p. 3].

Interestingly enough, the European Commission seems to be moving towards embracing software patents, which are considered to be "of importance for all enterprises in the software field, including [small to medium enterprises

(SME)][50, p. 12]. This is due to the recognition of the increasing importance of independent software developers and SMEs in spurring innovation and to the belief being adopted by European policymakers that:

> "Possession of [intellectual property rights (IPR)] helps any small company or individual independent software developer to raise finance to develop and market such inventions, and/or to license competitors and/or to sell or license his or her innovation to a major player. Possession of IPRs empowers the SME or individual. A patent is much more powerful in this respect than copyright[25, p. 2]."

Without the economic power to back up the anti-software patent position, policymakers are going to continue to embrace software patents, even to the OSS community's detriment. This is likely to be especially true if policymakers embrace the hypothesis that the OSS community is a gift culture, "an adaptation not to scarcity, but to abundance[53, p. 12]." If this hypothesis is embraced, then policymakers are likely to conclude that providing the best environment for economic activity will lead to a greater pool of developers who are in a position to be active OSS developers. From a policymaker perspective, the OSS community itself benefits more from having more developers, with a small proportion of OSS developers, than from having fewer developers, with a high proportion of OSS developers.

At least one study supports this position. The United States, which is the world's biggest software exporter, also produces the greatest number of OSS developers even though it has fewer OSS developers per capita than many countries[35].

3.2.2 Inconvenience

But perhaps the biggest reason why the opponents of software patents have not made any headway is that their arguments basically boil down to this: developers want to avoid inconvenience.

Developers have repeatedly argued that software patents will create a minefield of intellectual property that they will have to avoid. Yet, when a mine is hit, such as the LZW patent, they show examples of the innovation that policymakers want to see. For example, it seems unlikely that the PNG format, which is proclaimed as a better graphics file format than GIF, would have been developed if it was not for the LZW patent:

> PNG was designed to be the successor to the once-popular GIF format, which became decidedly less popular right around New Year's Day 1995 when Unisys and CompuServe suddenly announced that programs implementing GIF would require royalties[49].

Another argument made by OSS developers, the poor quality of patents, also boils down to a protest against inconvenience. What the OSS community does not seem to realize is that the strength of patents are widely varied. A patent that does not enable implementation of the invention can be thrown out. Even one example of prior art is enough to make a patent claim invalid. Legally savvy policymakers fail to see why such a fuss is being made over software patents in particular.

This issue of inconvenience due to patents exists in almost all fields. It is an issue that stems from the compromise policymakers have to make between a low cost patent examination process, which favors patenting by small entities, and a high quality patent examination process, which increases the cost of patent examinations.

Policymakers are willing to sacrifice the convenience of OSS developers for the sake of increasing intellectual property value, for providing an incentive for innovation.

4. WHAT THE OPEN SOURCE SOFTWARE COMMUNITY MUST DO

> *Hackers may cry about "freedom" and complain all they like about "corporate bean counters" taking away their "right" to appropriate others' patented technology, but they are doomed to eventual irrelevancy unless they come up with a strategy for dealing with the irreversible fact that patents are on the Net to stay.*
> *—Rembrandts in the Attic*[57, p. 193]

4.1 Understanding the system

Complaining, the only strategy that the OSS community is actively pursuing, is not going to do anything. Right now, members of the community show little understanding of what a patent actually covers. Headlines like "IBM Patents Web Page Templates[36]" on advocacy sites, and the ensuing comments, do little to support the anti-software patent stance. As Lee Hollaar, a Computer Science professor at the University of Utah, noted in his testimony in the 1994 Patent Hearings:

> Too often comments are based on the title or at best the abstract of the patent, and not the claims, which indicate what the true invention is. Often this is compounded by press releases from patentees trying to make their [patent] seem more important than it really may be[69, p. 32].

The scope of protection offered by a patent is much less than what is indicated by its title and abstract. First, a patent's claims limit the scope of what the invention covers. Second, a patent's claims must be examined in light of the actual invention disclosed[65]. Third, a patent's claims can be severely limited by the publicly available "prosecution history" of a patent, the concessions that were made by the patentee during the patent process in order to avoid having the claims cover any prior art (which would invalidate the claims).

OSS developers need to develop skill in reading patent claims and being able to tell if a product potentially infringes on them or not. Developing expertise within the OSS community on how to read the prosecution history of a patent, to clarify the scope of patent claims, would be extremely beneficial too.

The OSS community needs to develop the legal savvy needed to be effective against players in the software industry who might want to eliminate OSS competition, otherwise it will be possible for an aggressive player to use legal intimidation to shut the OSS community out of a market. Software does not even have to be patentable for the OSS community to suffer if the community cannot end the pattern of giving into legal intimidation.

4.1.1 Putting up a defense

OSS developers need to be prepared to put up a legal fight; otherwise, they have already lost.

Perhaps the best resource that the OSS community has is its incredible ability to mobilize members. Right now, a large number of the members are under-utilized.

Members of the OSS community could begin proactively creating a defense against threats of patent infringement suits. They could go through newly issued patents and published patent applications to classify what patent claims could be an issue for various OSS projects. Members could go through copies of the patent prosecution history and write up a response detailing why each claim is not infringed upon by potentially liable products.

OSS community members could also look for examples of prior art that could be used to invalidate potentially damaging patent claims. The rise of "Trash Patent Bins," such as IP.com, could help with this type of work.

Creating this kind of a library of information would give OSS developers a strong, inexpensive legal defense against claims of willful infringement. It would also reduce the legal costs and inconvenience in responding to a claim of patent infringement.

Most importantly, this kind of a project would lead to a better understanding of the patent system. OSS members would be more effective in their arguments with this kind of understanding.

4.1.2 Using antitrust and patent misuse

One area of law that the OSS community could leverage is antitrust and patent misuse law. Antitrust law, which is used to break up or regulate abusive monopolies, is often considered at odds with patent law, which grants legally protected monopolies. Both areas of law can be litigated privately and both strive to create the best environment for encouraging innovation. Patent misuse is another area of law that limits the ability of a patent holder to leverage a patent monopoly:

> Misuse involves some sort of licensing position that the courts have held to be out of sync with the patent grant. The defense departs from strict logic in that it would be incorrect to say: "I needn't license you at all; I can stand on my right to exclude. Therefore, I will do you a favor: I will license you *only on condition that* you purchase all your unpatented supplies from me; or take a license under several other patents; or agree not to deal in competitive goods; or agree to pay royalties based on your total sales of all goods, patented and unpatented; etc, etc." Such conditions, while seemingly implicitly lawful in light of the general power to exclude others from practicing the subject matter of a patent claim, are unlawful in cases where the patent carries significant market power to coerce covenants beyond its scope[31, p. 102]."

A patent holder that is found to have "unclean hands" under patent misuse law cannot enforce patent rights against an infringer until the misuse has been "purged."

The OSS community could use these bodies of law, in certain cases, to obtain licensing rights to key software patents. For example, it might be possible to build a successful case against Microsoft's refusal to license the patents needed to implement the CIFS using the CIFS Technical Reference to OSS developers licensing software under the GPL. The distribution of this software under their current RF license could be considered a form of intellectual property "dumping" designed specifically to target Samba in a discriminatory fashion.

It is possible that Microsoft could find a way of arguing that allowing licensees to use Microsoft's related patents on a RF basis, yet not allowing licensees to sublicense these same patents on a RF basis, is somehow procompetitive. After all, even the DOJ's recent antitrust settlement with Microsoft specifically did not require Microsoft to allow its patents to be sublicensed[60, p. 6]. However, this license could encourage the development of commercial implementations of CIFS, which could crowd out Samba and Linux. At the same time, the Samba team could find itself unable to create compatible software without infringing on Microsoft's patents. At any time, Microsoft could release a new version of CIFS, one with new software patents licensed under RAND terms that target whichever competing product takes the place of Samba. With just a little maneuvering, Microsoft could always license the latest CIFS patents as RAND, yet still create an anticompetitive atmosphere. Thus, there is a strong argument that the anticompetitive effects of Microsoft's refusing to license its patents for use under the GPL are greater than the procompetitive effects of the RAND licensing of these same patents.

Making these kinds of arguments and effectively using them in court is not easy—or cheap. However, an accused patent infringer has remedies, including punitive damages if threatened with a patent infringement lawsuit that is filed with "bad faith[31, pp. 79–122]."

Unfortunately, antitrust law only works against companies with market power. Microsoft is a relatively easy target because it has legally been branded as an aggressive monopoly[30]. However, any company or individual can legally obtain a software patent and have the potential power to limit the kind of software that the OSS community can develop, yet not every company or individual will be assumed to have market power because of an issued patent. Patent misuse, though, does not depend on the patent holder having market power.

Antitrust and misuse law could be useful in some cases, but the OSS community should not depend on these bodies of law to guarantee unlimited freedom to develop software. The OSS community should find ways of avoiding litigation in the first place.

4.1.3 Best mode

There are other aspects of patent law that the OSS community could exploit. One interesting example is the "best mode" requirement:

> [The best mode requirement is] a condition of the grant of a valid patent, under which the inventor is required to describe the best method known to the inventor of carrying out the claimed invention. The inventor must not conceal from the public the best physical way to make use of the invention[37, p. 38]

Argued effectively, this requirement could be very devastating to the intellectual property value of many software patents, particularly those that cover changes to de facto standards. For example, a case could be built around the best mode requirement to have two patents covering Microsoft's VFAT file system thrown out because they refer to a "checksum" field, but do not disclose the best checksum algorithm to use[55, 56]. The only reason one would ever use these two patents would be to implement Microsoft's VFAT file system, so the only reasonable checksum algorithm to use, the best checksum algorithm to use, would be the checksum algorithm used by Microsoft. The OSS community could rid itself of various software patent nuisances such as these by arguing some of these cases.

Not all software patents would crumble under the best mode requirement, but it is a good example of an aspect of patent law that, better understood by the OSS community, could be used to protect the OSS community from software patents.

4.2 Sharing some lessons with the USPTO

Members of the OSS community could become more effective participants in issues surrounding software patents by changing the focus of the their arguments. Arguing against the patentability of software has been counter-productive. For example, focusing on this issue has led to a "lower level of scrutiny regarding novelty and nonobviousness[27, §III]" in the USPTO.

There has been more and more debate focusing on these last two issues. But right now, the only actions that one can take with regards to a patent issued that does not meet requirements of novelty and nonobviousness are to a) request a reexamination, b) litigate for a declaration of non-infringement, c) ignore the patent, or d) remove products from the market that one thinks may infringe on the patent. The OSS community has consistently chosen one of the last two options.

Part of the problem is the delay between when a patent application is filed and when a patent is issued. By the time a patent is issued, many OSS projects may already have incorporated the technology, unaware of the future patent pitfall. The OSS community would benefit if it successfully lobbied the USPTO to adopt the "release early, release often[52, p. 7]" philosophy. Currently, patents take three years to be issued. It would be a great service if they were released in intermediate forms during the course of prosecution. Allowing patent holders to exercise intellectual property rights early in a weaker form, with a lower level of deference to the validity of claims, would benefit both the patent holders, who could negotiate using the patents more quickly, and possible infringers, who could avoid implementing infringing software. This early-release approach to software patents could even been seen as beneficial from a policymaker's perspective because early release could result in more participation in the patent process by individuals and other small entities.

The OSS community would also benefit if it helped get the USPTO even more online. Today, any patent can be viewed online. However, one must still go to USPTO to get copies of a patent's prosecution history, or pay services to make copies of these histories. If the USPTO were to make these prosecution histories available online right away, then the public would benefit from increased access to these histories and the USPTO would benefit from an improved perception of the patent examination corps' work.

4.3 Embracing the system

The OSS community has benefited greatly when its members have been willing to make compromises.

Recently, the coinage of "open source," an alternative to the less commercial-friendly term, "free software," shows that the OSS can benefit from compromise. Part of the incredible investment made into OSS may stem from this approach, which is less adversarial than the one taken by the Free Software Foundation.

At some level, the OSS community must compromise the community's stance against the patent system in order to minimize the risk that software patents pose to OSS. The OSS community must embrace this form of intellectual property.

4.3.1 An old idea

The best example of the OSS community embracing an intellectual property system to push an alternative approach to intellectual property is the formation of the "copyleft," a reaction by the Free Software Foundation to the restrictions that can be placed on users because of copyright law. The "copyleft," embodied by the GNU General Public License, leverages copyrights to ensure that intellectual property created by developers who license software under the GPL do not lose their intellectual property to proprietary software developers without the OSS community receiving compensation in the form of more intellectual property; that such software remains "free."

The OSS community should embrace a more powerful form of intellectual property, software patents, not only to protect itself, but also to help it flourish. The OSS community should form an open source patent pool.

Forming an OSS patent pool is not a new idea. The League for Programming Freedom has long suggested that developers should adopt a "nonaggression or mutual defense policy[44]." Karsten Self, a consultant for Kuro5hin.org, even proposed an OSS patent pool in 1998[61], developing some interesting ideas. Others have also brought up the possibility of creating an OSS patent pool. However, interest in pursuing the development of one seems to have died down, perhaps because the question of how to make an OSS patent pool flourish has not been directly addressed.

5. FINDING THE COMMON GROUND

What an OSS pool needs to do to flourish is to find the common ground shared by the OSS community, the proprietary software industry, and the U.S. government.

5.1 Taking the moderate approach with Reasonable and Non-Discriminatory

An OSS pool would find this common ground by sublicensing a large body of patents under RAND terms. All the OSS community needs to do is define "for free use in open source software" as RAND. This would create an incentive for the disclosure of innovations.

This incentive to innovate would come from contracts with organizations guaranteeing the pool the right to sublicense patents for open source software. This would leverage the power of the OSS culture because developers, unwilling to disclose inventions out of concern for the health of the OSS community, would then be willing to participate in the patent process.

Another incentive to innovate would come from the increased value that software patents would have for small entities. An OSS pool would not only pool software patents, but also the resources needed to seek out infringers. It could even pool together the resources needed for small entities to be able to successfully sue infringers with deep pockets who refuse to license the software patents of the small entities.

However, sublicensing patents for use in open source software is not useful by itself. Individual patent holders could do the same thing and the effect would not differ. At least one patent is already licensed under this type of license[47] and another OSS license has been proposed to "promote the development and use of Open Source software[22]," but licensing patents this way provides no leverage to the OSS community as a whole.

However, an OSS patent pool that had the right to sublicense some of the patents for proprietary use would be able to leverage the patents on behalf of the OSS community. Having this right would mean that that OSS community could use the patents as bargaining chips when dealing with those exercising their own patent rights.

Being a point of negotiation for the proprietary use of some software patents would also help the OSS patent pool itself grow. Organizations could use the patent pool as a way of exploiting some of their "non-core" software patents. The royalties from these software patents would benefit the OSS patent pool, which would take a portion to cover administrative costs, and the organizations licensing the patents; intellectual property that would otherwise just result in costly maintenance fees. Accumulated royalties themselves might be an important bargaining chip for the OSS community too since some holders of important patents would be willing to settle any infringement suits against OSS developers with a one-time payment.

5.2 Avoiding the antitrust risk

An OSS patent pool could probably successfully defend itself from any antitrust suit by only licensing software patents non-exclusively. This would avoid antitrust action because, if a party did not want to license patents from the OSS patent pool, then the party could always go to the individual patent asignees. The situation would be no different than if the OSS patent pool did not exist at all.

The OSS patent pool would still become the point of negotiation for many patents. Many patent holders would likely refuse to license directly to other developers because of the administrative costs and because the OSS patent pool would be unlikely to look for infringers of patents that the OSS patent pool would effectively be unable to use as bargaining chips.

Unfortunately, an OSS patent pool could be accused of "effectively" licensing software patents on an exclusive basis, even if all patent licensing contracts were non-exclusive. To ensure that an OSS patent pool could avoid being destroyed for violating antitrust law, the argument would have to be made that an OSS patent pool would have procompetitive benefits—that an OSS patent pool would create an incentive for innovation.

Part of this incentive could come from claiming that protecting the OSS market itself protects an environment for innovation. This argument would depend on the ability to successfully argue that OSS development does inspire innovation in ways that commercial software development does not. Another part of the argument that an OSS patent pool would create an incentive for innovation would come from claiming that such a pool would result in increased disclosure of innovations from participating individuals and organizations.

Fortunately, it is unlikely that the proposed strategy to licensing patents would cause much antitrust concern. Non-exclusive licensing of patents, and sublicensing them on a RAND basis, would mean that the pool could not be easily accused of cartel activity. This strategy also avoids forcing anyone to license software patents to proprietary competitors, or to only license through the OSS patent pool, thus it avoids diminishing the incentive to patent.

One weak point of an OSS patent pool could come from the pool's indiscriminate licensing of software patents. Most patent pools are used to license complementary patents as a package. Yet, it makes little sense to have multiple OSS patent pools all sublicensing patents for use in OSS.

The OSS patent pool could also come under fire for the proprietary licensing of patents, an aspect that would make the OSS patent pool act more like a patent holding company than a traditional pool. Non-exclusive licensing of these patents by the OSS patent pool would strengthen the pool's position, but it would be wise for the OSS patent pool to continually build up an argument that its actions and structure were procompetitive.

5.3 Acquiring the patents

For an OSS patent pool to have an effect, it must have patents. However, there has been little OSS patenting activity. Fortunately, a small number of patents licensed to an OSS patent pool, along with the transferrable right to sublicense them to proprietary software developers, would be enough for the pool to raise funds and begin a campaign to acquire the rights for more patents. At that point, an OSS patent pool would have many opportunities for acquiring rights to sublicense software patents.

One method would be to acquire software patent sublicensing rights during negotiations for proprietary licensing of existing patents in the pool. This would be the method that could best be used to turn software patents into a defense, especially if the OSS community mobilized to actually identify what software patents could cause the most damage to the community.

Another method would be to acquire software patents through "fire sales." The software industry is an incredibly dynamic industry and the OSS patent pool could acquire software patents from companies going out of business. These software patents could be transferred back to the original inventors, who could then license the patents back to the OSS pool. Had an OSS pool been established before the bursting of the "dot bomb" stock bubble, it could have already acquired the rights to a great number of patents this way.

One method that could risk alienating potential software patent licensors, but that might be worthwhile, would be to go after holders of weak patents. Software patent holders would likely rather see their patents licensed to the OSS community rather than thrown out altogether. This strategy has already been used successfully in the commercial world by Barr Laboratories, which "has earned $200 million in recent years by challenging patents it considers 'breakable' in court and forcing their owners to either supply their patent drugs to Barr for generic sale (such as Zeneca Group's Tamoxifen) or else pay annual fees to make Barr go away (such as the $30 million a year Bayer will pay until the year 2003)[57, p. 48]."

An OSS patent pool could also license its patents directly from individual OSS developers, something that policymakers would be very happy to see happen. An OSS patent pool could offer many incentives for OSS developers, aside from the benefits to the OSS community itself, to take on the patent system and acquire their own software patents. Developers could be offered royalties, patent defense insurance and even patent prosecution assistance.

No matter how an OSS patent pool acquires its patents, the benefit of working with the patent system would be tremendous. Creating a successful OSS patent pool would take a lot of work, but the result could be as effective for the OSS community as the nineteenth century's railroad associations, which "virtually removed patent rights from consideration in railroad innovation[70, p. 204]."

6. CONCLUSION

The OSS community faces a very real threat from software patents. Software patents have been legitimized in the legal and business world and the OSS community will be unlikely to turn this around. Now the OSS community needs to develop a greater degree of legal savvy if OSS is to be able to keep up with the innovations in proprietary software. By using the patent system to create an OSS patent pool, the OSS community would develop this savvy and help ensure the survival of OSS.

7. ACKNOWLEDGEMENTS

This paper could not have been written without the consistent support of my thesis advisor, Professor Lee Hollaar. He is the source of some of the best ideas in this thesis. His perspective of these issues as an insider has made his advice invaluable to me. His ability to strip complicated issues down to a simple core issue has been inspirational.

Another person who made this paper possible was my undergraduate advisor, Sandy Hiskey. She always believed that I would finish this paper and graduate—even when I began to think that I would not.

While I am at it, I would like to thank Professor Joel Dubow. I do not even know if I would have made it this far if he had not given me the encouragement I needed to get into the University of Utah's Computer Science program.

8. REFERENCES

[1] 1966 Report of the President's Commission on the Patent System. 1. Cited in [66, pp. 24–26].

[2] 35 U.S.C. §101.

[3] Aharonian, G. 1995 software patent statistics. *Internet Patent News Service*. `http://lpf.ai.mit.edu/Patents/1995-patents.html`.

[4] Apparatus and method for compressing data signals and restoring the compressed data signals. U.S. Patent No. 4,464,650, 1981.

[5] Battilana, M. C. The gif controversy: A software developer's perspective. `http://lzw.info`, 1995.

[6] Berman, H. L. Statement of rep. howard l. berman: Introduction of h.r. 5364, the business method patent improvement act of 2000, Oct. 3, 2000.

[7] Boutell, T. Gd 1.8.4 manual. `http://www.boutell.com/gd/manual1.8.4.html`.

[8] Burn all GIFs. `http://burnallgifs.org/`.

[9] ComputerWire. Microsoft anti-GPL fine print threatens competition. `http://www.theregister.co.uk/content/4/24885.html`, Apr. 17, 2002.

[10] DeMocker, J. Tetris pressures game act-alikes. *Wired News* (Dec. 4, 1998). `http://www.wired.com/news/culture/0,1284,16619,00.html`.

[11] Diamond v. Chakrabarty. 447 U.S. 303 (1980). Cited in [66, p. 30].

[12] Diamond v. Diehr. 450 U.S. 175 (1981). Cited in [66, p. 30].

[13] Durand, D. G. Eliminate RAND option from new patent policy, only allow RF. W3C www-patentpolicy-comment mailing list. `http://lists.w3.org/Archives/Public/www-patentpolicy-comment/2001Oct/0737.html`.

[14] Economic report of the President. `http://w3.access.gpo.gov/usbudget/fy2002/pdf/2001_erp.pdf`, Jan. 2001.

[15] Examination guidelines for computer-related inventions. Patent and Trademark Office, Department of Commerce. `http://www.uspto.gov/web/offices/pac/dapp/oppd/pdf/ciig.pdf`.

[16] Farrell, J., and Saloner, G. Coordination through committees and markets. *RAND Journal of Economics 19*, 2 (Summer 1988), 235–252. `http://links.jstor.org/sici?sici=0741-6261%28198822%2919%3A2%3C235%3ACT%CAM%3E2.0.CO%3B2-U`.

[17] Gable, R. L., and Dengler, A. B. What can practitioners do in light of Alappat, Lowry, Trovato and similar cases? `http://web.idirect.com/~dcameron/gabl0001.htm`, 1995.

[18] Gilbert, R., and Shapiro, C. Antitrust issues in the licensing of intellectual property: The nine no-no's meet the nineties. *Brookings Papers on Economic Activity* (1997), 283–336.

[19] GNU General Public License. `http://www.gnu.org/copyleft/gpl.html`.

[20] GNU Lesser General Public License. `http://www.gnu.org/copyleft/lesser.html`.

[21] The GNU manifesto. `http://www.gnu.org/gnu/manifesto.html`.

[22] Goals for the opl. `http://www.openpatents.org/license/goals.php`.

[23] Gottschalk v. Benson. 409 U.S. 63 (1972). Cited in [66, p. 134].

[24] Harry Wolin, E. Standard setting in the semiconductor industry: Trends, benefits and anti-competitive concerns. FTC/DOJ Hearings on Competition and Intellectual Property Law and Policy in the Knowledge-Based Economy, Mar. 20, 2002. `http://www.ftc.gov/opp/intellect/`

020320harrywolin.pdf.

[25] Hart, R., Holmes, P., and Reid, J. The economic impact of patentability of computer programs. http://europa.eu.int/comm/internal_market/en/indprop/comp/study.pdf, Dec. 15, 2000.

[26] History of the league for programming freedom. http://lpf.ai.mit.edu/History/history.html.

[27] Hollaar, L. A. Justic Douglas was right. *AIPLA Quarterly Journal 24*, 1 (Winter 1996), 283–305. http://www.cs.utah.edu/~hollaar/aipla-qj.html.

[28] H.R. 5364. 106th Congress 2nd Session, Oct. 3, 2000.

[29] Html 4.01 specification. http://www.w3.org/TR/html401/, Dec. 24, 1999.

[30] Jackson, T. P. Findings of facts. United States of America v. Microsoft Corporation, C.A. 98-1232, Dec. 21, 1999. http://usvms.gpo.gov/findings_index.html.

[31] Janicke, P. M. *Modern Patent Litigation*. Carolina Academic Press, Durham, North Carolina, 1999.

[32] Katz, M. L., and Shapiro, C. Antitrust in software markets. http://faculty.haas.berkeley.edu/shapiro/software.pdf.

[33] Klein, J. I. Acting Assistant Attorney General Antitrust Division U.S. Department of Justice. Cross-licensing and antitrust law address before the American Intellectual Property Law Association, May 2, 1997. http://www.usdoj.gov/atr/public/speeches/1123.htm.

[34] Krechmer, K. Communications standards and patent rights: Conflict or coordination? *Telecommunications Industry Association's STAR* (1997).

[35] Lancashire, D. The fading altruism of open source development. *First Monday 6* (Dec. 2001). http://firstmonday.org/issues/issue6_12/lancashire/index.html.

[36] Lefkowitz, J. IBM patents web page templates. http://slashdot.org/article.pl?sid=01/10/17/005232.

[37] McCarthy, J. T. *McCarthy's Desk Encyclopedia of Intellectual Property*, 2 ed. BNA Books, Washington, D.C., 1995.

[38] Memorandum in support of commissioner's motion to dismiss. Cited in [17, §VII.C].

[39] Method and system for providing a common name space for long and short file names in an operating system. U.S. Patent No. 6,286,013 B1, 2001.

[40] Miano, J. *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP*. Addison Wesley Longman, Inc., Reading, Massachusetts, 1999.

[41] Miller, R. Unisys not suing (most) webmasters for using GIFs. http://slashdot.org/article.pl?sid=99/08/31/0143246, Aug. 31, 1999.

[42] Mitchell, R. Open war. *Wired* (Oct. 2001). http://www.wired.com/wired/archive/9.10/linux_pr.html.

[43] Morgan, J. V. Chaining open source software: The case against software patents. http://lpf.ai.mit.edu/Patents/chaining-oss.html, 1999.

[44] Mutual defense against software patents. League for Programming Freedom, Jan. 28, 1994.

[45] Nader, R. Error message: Microsoft has performed an illegal function and should be shut down. *In the Public Interest*, May 8, 2000. http://www.sfbg.com/nader/100.html.

[46] North America's top 100 computer companies. http://www.netvalley.com/top100am_vendors.html. Cited on May 6, 2002.

[47] The open rtlinux patent. Finite State Machine Labs. http://www.fsmlabs.com/openpatentlicense.html.

[48] Open source definition. http://www.opensource.org/docs/definition.html.

[49] PNG: What it be (an informal history). http://www.libpng.org/pub/png/#history.

[50] Proposal for a directive of the European Parliament and of the council on the patentability of computer-implemented inventions. http://europa.eu.int/comm/internal_market/en/indprop/com02-92en.pdf, Feb. 20, 2002.

[51] Public issues for patent policy framework of 20010816. W3C. http://www.w3.org/2001/11/PPF-Public-Issues.html.

[52] Raymond, E. S. The cathedral and the bazaar. http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/cathedral-bazaar.ps, 2000.

[53] Raymond, E. S. Homesteading the noosphere. http://www.tuxedo.org/~esr/writings/homesteading/homesteading/homesteading.ps, 2000.

[54] The results of the European Comission consultation exercise on the patentability of computer implemented inventions. Final Report by PbT Consultants.

[55] Reynolds, A. R., et al. Common name space for long and short filenames. U.S. Patent No. 5,578,517, 1996.

[56] Reynolds, A. R., et al. Method and system for providing a common name space for long and short file names in an operating system. U.S. Patent No. 6,286,013 B1, 1997.

[57] Rivette, K. G., and Kline, D. *Rembrandts in the Attic: Unlocking the Hidden Value of Patents*. Harvard Business School Press, Boston, Massachusetts, 2000.

[58] Roelofs, G. Moderator Q&A. http://www.sonic.net/~roelofs/reports/linux-19980714-phq.html, July 19, 1998. From an informal report on "The Future of Linux" panel.

[59] Royalty-free CIFS technical reference license agreement. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnkerb/html/Finalcifs_LicenseAgrmnt_032802.asp.

[60] Second revised proposed final judgement. United States of America v. Microsoft Corporation, C.A. 98-1232. http://www.usdoj.gov/atr/cases/f10100/10146a.pdf. This is Microsoft's antitrust settlement with the DOJ and nine States.

[61] Self, K. M. Cooperative oss patent pool—proposal. http://home.netcom.com/~kmself/OSSPatPool/

index.html, 1998.

[62] Shapiro, C. Competition policy in the information economy. http://faculty.haas.berkeley.edu/shapiro/comppolicy.pdf, Aug. 1999.

[63] Shapiro, C. Navigating the patent thicket: Cross licenses, patent pools, and standard-setting. University of California at Berkeley, Mar. 2001. http://haas.berkeley.edu/~shapro/thicket.pdf.

[64] Shared source vs. open source: Panel discussion. http://linux.oreillynet.com/pub/a/linux/2001/08/09/oscon_panel.html?page=1.

[65] Silverman, A. B. Claim interpretation: Do not be misled by a means for recital. *JOM: The Member Journal of The Minerals, Metals & Materials Society 53*, 4 (2001), 48. http://www.tms.org/pubs/journals/JOM/matters/matters-0104.html.

[66] Stobbs, G. A. *Software Patents*, second ed. Aspen Law & Business, A Division of Aspen Publishers, Inc., Gaithersburg, New York.

[67] Technology assesment and forecast report: Electrical classes. U.S. Patent and Tradmark Office, May 2001. http://www.uspto.gov/web/offices/ac/ido/oeip/taf/stelec.pdf.

[68] The Unisys/CompuServe GIF controversy. http://lpf.ai.mit.edu/Patents/Gif/Gif.html.

[69] United States Patent and Trademark Office: Public hearings on use of the patent system to protect software-related inventions, Jan. 26–27, 1994.

[70] Usselman, S. W. Organizing a market for technological innovation: Patent pools and patent politics on American railroads, 1860–1900. *Business and Economic History, second series 19* (1990), 203–212.

[71] Valloppillil, V., and Cohen, J. Linux OS competitive analysis: The next Java VM? http://www.opensource.org/halloween/halloween2.html, Aug. 1998.

[72] Web site LZW licenses available from Unisys. http://www.unisys.com/unisys/lzw/lzw-license.asp.

[73] Weitzner, D. J., e. W3C patent policy framework: W3C working draft. http://www.w3.org/TR/2001/WD-patent-policy-20010816/, Aug. 16, 2001.

[74] Weitzner, D. J., e. Patent policy working group royalty-free patent policy: W3C working draft. http://www.w3.org/TR/2002/WD-patent-policy-20020226/, Feb. 26, 2002.

[75] What's Samba all about? http://www.samba.org/samba/about.html.

[76] Why no GIFs. http://www.gnu.org/philosophy/gif.html.