Safe harbors and F/OSS community interests: A new approach to United States patent reform advocacy

Jason V. Morgan, Patent Attorney

J.D. 2008, George Washington University Law School
B.S. Computer Science 2002, University of Utah

Abstract. Software patents have been a concern in the Free/Open Source Software (F/OSS) community for many years. Some F/OSS supporters have been very vocal in their opposition to software patents, yet software patents have a solid foothold in United States patent law. Patent law continues to evolve, but F/OSS interests are not being addressed directly. The focus of some F/OSS advocates on the abolition of software patents has been to extreme a position to move United States policymakers to act. Thus, F/OSS interests are only advanced to the extent that they coincide with the interests of the broader software industry. This is problematic; F/OSS interests do not strictly coincide with the interests of the broader software industry. F/OSS advocates can create a new balance of power by focusing on F/OSS community interests rather than holding out for an end to software patents. Focusing on the interests opens up potential, modest patent reforms. One set of modest reforms are safe habor protections, laws establishing rules that, if followed, exempt parties from key legal consequences. Introducing a series of safe harbors to allow F/OSS contributors to act without fear of taking on unmanageable patent infringement risks could be as effective as eliminating software patents when it comes to protecting F/OSS community interests. With policymakers struggling to adapt the patent system to the competing needs of the pharmaceutical and software industries, F/OSS advocates could obtain safe habors as concessions from policymakers seeking support for other patent reforms. Acting quickly to secure safe harbor protections could neutralize the most significant threats that the patent system could pose for F/OSS.

1 Introduction

Software Software patents are a great concern to members of the Free/Open Source Software (F/OSS) community. Software patents are potentially very powerful litigation tools and F/OSS community members usually seek to avoid litigation. While the impact of software patents on F/OSS efforts has been rather limited so far, many risk-averse F/OSS contributors could abandon their efforts were software patent holders to begin asserting their claims more aggressively. This would dampen F/OSS efforts, depriving the F/OSS community the benefits of shared knowledge and skill. It would also deprive F/OSS consumers the benefit of a large pool of actively maintained software.

For many years, F/OSS advocates have sought to address these risks by attacking the very foundations of software patentability. These efforts have had some effect in stimulating software patent angst; however, they have not led to the destruction of

software patents. The evolution of patent law has weakened much of the potential impact of software patents, but the potential threat to the F/OSS community remains.

Continuing to fight against software patents as a matter of principal is likely to lead to many more years of frustration. Another approach to software patent risk management is needed. Engaging in principled negotiation, the subject of the seminal work, Getting to Yes[26], could help the F/OSS community make quick progress to defuse the potential software patents time bomb. With the United States Congress engaged in a tiring effort to reach agreement on other patent reforms, F/OSS advocates could secure significant concessions that would protect F/OSS interests for years to come.

2 The F/OSS community should seek safe harbor protections

There are many concessions other than the end of software patents that F/OSS advocates can fight for. Perhaps the most significant set of concessions would be changes in the United States patent code to offer the F/OSS community a series of targeted safe harbors—areas of law that protect one from certain liabilities, just like a ship that is docked in a safe harbor is protected from the dangers of the sea[34].

Safe harbors that would protect the F/OSS community from most patent infringement squalls would focus on protecting F/OSS contributors and non-profit F/OSS distributors from damages and injunctions. Individual F/OSS contributors, and even non-profit F/OSS distributors, are the most vulnerable to software patent infringement claims. It is best to distinguish them from commercial F/OSS distributors, who are generally better positioned to engage in legal battle.

First, F/OSS contributors should be protected from individual monetary or injunctive relief for their efforts in developing and testing software that may infringe on patents. While patent suits against individuals are rare, "whoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent[5]." No safe harbors would be workable if they neglect the process of creating potentially infringing software.

F/OSS contributors and non-profit distributors should be protected from monetary damages for patent infringement claims if they include documentation allegations of infringement with allegedly infringing F/OSS products. The actual burden of identifying whether a patent might be infringed by a given F/OSS product should be borne by the software patent holders themselves.

Monetary damages would not go away altogether. Commercial F/OSS distributors and F/OSS users could still be liable. But they could decide for themselves whether to risk infringement, whether to negotiate with software patent holders for separate licenses, or whether to modify the F/OSS to avoid infringement. Some commercial F/OSS distributors might decide to indemnify their users from patent liability. Commercial F/OSS distributors might even negotiate with software patent holders on behalf of their users, although some F/OSS licenses could hinder such efforts[28].

Third, all F/OSS contributors and distributors should be free from injunctive relief for F/OSS products that do not enable infringing behavior by default. That is, if an F/OSS product requires the user to activate infringing functionality, then a court should not be allowed to order F/OSS contributors and distributors to stop distributing the software.

For a patent holder, disabling the infringing software solution would, in most cases, be nearly as effective as an injunction preventing the distribution of the software. Most users would never even consider turning on the allegedly infringing features. Not all users who did consider turning on such features would do so, especially since they might face increased damages if they are found to have engaged in willful infringement[7].

Methods of automating the enablement of infringing behaviors could be devised so that those who have licensed particular patents—or believe them to be invalid or not infringed—would be able to use allegedly infringing solutions with minimal inconvenience. Those using the software outside of the United States would also benefit from the automatic enablement of all features that infringe only United States patents, which have no legal effect overseas.

And finally, ideal safe harbor provisions would ensure that, if disabling certain features also disabled substantial non-infringing behavior too, then F/OSS contributors and distributors would be free from injunctive relief without having to disable those features by default. Of course, a patent holder could still obtain injunctive relief by proving that the infringing behavior could be disabled by default without disabling substantial non-infringing behavior. But proving this would probably entail actually writing appropriate code, which could then be incorporated into the infringing F/OSS product. Thus, the burden of writing code to disable infringing behavior in F/OSS products would probably fall on patent holders themselves, not F/OSS contributors.

3 The Digital Millennium Copyright Act shows that safe harbors can be effective instruments

The proposed F/OSS safe harbors would significantly reduce the possible effects software patents might have on the F/OSS community. They are, in fact, similar to some of the safe harbor provisions found in The Digital Millennium Copyright Act of 1998 (DMCA)[62]. Specifically, they are most like the notice-and-takedown / counternotice-and-putback procedures[2].

These procedures shield service providers from monetary damages for copyright infringement based on data uploaded by users. To take advantage of this safe harbor, a service provider must designate an agent to receive notice of copyright infringement claims. When a copyright holder provides proper notice to the agent of allegedly infringing activities, the service provider must remove or disable access to the allegedly infringing content.

To avoid potential liability to the user whose content was removed, the service provider must notify the user of the notice and takedown. If the user provides proper

notice that the content should not have been removed, then the service provider must put the material back online. The service provider must then put the content back online.

Following these basic steps protects the service provider from the risk of having to pay monetary damages. Such risk is shifted to the users, who can fight the copyright holders if they desire. Moreover, the burden of policing user-provided content is shifted to the copyright holders. While this may not be as efficient as having the service providers policing content, it certainly is less inhibiting. Service providers do not have to keep track of which copyright holders let their works fly free and which copyright holders keep a tight reign on the content. Service providers can just wait for copyright holders to point out problems and then react accordingly.

It is undisputable that there are those who would abuse the notice-and-takedown procedure based on flimsy claims[79]. Fortunately, the counternotice-and-putback procedure provides a nice tool for countering such abuses. This is an important feature that is lacking in the European Union version of this safe harbor; the European Union version only includes notice-and-takedown, but does not include counternotice-and-putback[23].

Service providers themselves can, if they chose, stand up against abusive practices of content holders. There is no mandate that service providers stay within the safe harbor. The United States Congress made it clear that other defenses were not to be affected by the introduction of safe harbors[1]. Thus, just like a ship can head to open waters, even when dark clouds loom, service providers with reasonable risk appetite can venture out of their DMCA safe harbors to fight for the rights of their users.

Not all service providers are willing to take on the costs and risks involved in fighting for the rights of their users. But, those same service providers who take every takedown notice at face value would probably not have been willing to stand up for their users were the DMCA to have never passed. Moreover, many service providers might have started to actively police user postings, thus stifling freedom on the Internet without giving the users the option of having their day in court.

The notice-and-takedown / counternotice-and-putback safe harbor of the DMCA has successfully given risk-averse service providers the option of staying out of the way of content control battles. To date, over nine thousand service providers have registered designated agents with the United States Copyright Office[16]. Service providers set to take advantage of this safe harbor host over forty thousand sites, including well-known sites such as YouTube, Slashdot.org, Craigslist, Wikipedia, and SourceForge. Yet, many, many more service providers have chosen not to designate an agent. Presumably, these service providers either do not host significant user content or they are willing to defend the rights of their users.

The notice-and-takedown / counternotice-and-putback safe harbor provision of the DMCA strikes a nice balance among the interests of copyright holders, service providers, and users. Copyright holders can quickly have infringing materials removed from sites, service providers do not have to police their sites for infringing material and they do not have to take on unwanted risk, and users can decide for themselves if they have rights worth fighting for.

Safe harbors can work well when they shift the risks and burdens away from parties with weak interests to those parties with strong interests—interests that they are willing to actively protect. The F/OSS community should seek safe harbors to shift the risks and burdens associated with patent law to the parties most willing to protect their interests.

4 Patent reform is needed to protect the F/OSS community from the negative effects of software patents

The United States Congress is unlikely to engineer safe harbors to protect the F/OSS community unless advocates show that the F/OSS community is worth protecting and that the F/OSS community is vulnerable.

Fortunately, the evidence is out there that the F/OSS community provides a valuable service by producing and distributing a variety of software products that users are allowed to modify and redistribute[20, 50]. Billions of dollars in information technology costs have been saved as a result of F/OSS community efforts[72]. Models of F/OSS usage suggest that much of this money is invested, through mechanisms such as in house and contract software development, to create productivity-enhancing technologies[60]. Custom software engineering firms have certainly found great value in F/OSS, which has provided such firms with cost-effective, reliable, interoperable, customizable toolsets that often come with built-in support communities[52].

There is also evidence that software patents pose a potential threat to the emerging F/OSS phenomena [50]. Patents are a very powerful form of intellectual property. A valid patent grants its holder the right to exclude others from making, using, offering to sell, selling, or importing the invention claimed in the patent[5]. This right can be enforced in the courts through injunctions and the award of damages to the patent holder[6, 7]. Under exceptional circumstances, courts may even make a patent infringement defendant pay the patent holder's attorney fees[8].

F/OSS community participants typically have limited access to resources with which to fight claims of patent infringement, let alone with which to pay off successful infringement claims. They face potentially great exposure in developing and distributing technology that may infringe on patent holder claims. Thus, F/OSS community participants, like the participants in comparable software development communities, are generally risk-averse.

With software patent holders showing signs of increasing aggression, many F/OSS community participants could be dissuaded from their efforts. Safe harbor protections would protect the future of F/OSS.

4.1 Participants of the F/OSS community and comparable communities are generally risk-averse

Decision makers might question whether existing laws are sufficient to protect F/OSS interests. After all, F/OSS contributors can try to avoid patent infringement; they can also defend themselves in court if they are not actually infringing. But, F/OSS contributors are generally not the parties best suited to fight patent infringement suits. They are not writing code to make points about patent law. Generally, they just want to share their work with each other.

When intellectual property rights claims are asserted, F/OSS community participants often avoid the technologies at issue rather than take on the cost of defending their position. For example, the popular GD graphics library dropped support for producing GIF files for a period of time, likely because of the aggressiveness of Unisys in asserting its patent rights over the Lempel-Ziv-Welch (LZW) compression algorithm[51]. The GD code did not even use the LZW algorithm; a workaround algorithm had been developed that created compatible, although somewhat larger, GIF files[50].

Even after the much-despised LZW patent[75] expired, many F/OSS advocates, such as the Free Software Foundation and GNU, refused to use the LZW algorithm because of another patent[76], assigned to IBM, which appeared to also claim the LZW algorithm[80]. Instead of pushing IBM to dedicate its patent to rights to the public, and instead of mounting an argument that the patent was invalid because it was anticipated by or obvious in light of prior art[3, 4], the F/OSS community simply avoided LZW technology altogether.

Risk-averse behavior among members of the F/OSS community and comparable communities is evident with regards to other intellectual property laws too. The apparent power wielded by The Tetris Company against developers of Tetris clones is a notorious example. The Tetris Company has managed to suppress clones of the classic game of Tetris, using little more than a shaky copyright claim[51] and the (easily avoidable) claim of trademark infringement. Noah Witherspoon, creator of Tris, the non-F/OSS, but freely-available remake of Tetris, is the latest person to pull a clone of the twenty-three year old game[71]. Mr. Witherspoon is "a college student, and not an affluent one, and [he] simply do[es] not have the time, energy, or resources to fight this battle right now[81]." So far, The Tetris Company has exercised patent-level protection through bluster alone: the company has never even filed suit in a United States district court[17].

Of course, some F/OSS contributors may be willing to take measured risks. One bright example of measured risk-taking is being exhibited by Professor Robert Jacobsen, the Berkeley physics professor who has contributed much to model railroad enthusiasts through his efforts with the Java Model Railroad Interface (JMRI) project[39]. In 2005, a lawyer for KAM Industries (KAM) sent Professor Jacobsen a letter stating their belief that the JMRI software infringed one of KAM's patents[78] and asking Professor Jacobsen to rewrite the JMRI code or license the patent at $19 per copy of

the software[40]. Dr. Jacobsen did neither. He stood his ground, going so far as to file suit seeking to have the patent declared invalid [38].

The F/OSS community probably has its fair share of Jacobsens, individuals who, if pushed up against the wall of alleged patent infringement, will strike back with ferocity. Supporters of individual efforts may even provide financial backing in many instances, adding strength to retaliatory efforts. But many, if not most, F/OSS contributors are more like Mr. Witherspoon. They are willing to expend great energy as developers, but do not feel comfortable making a stand against intellectual property saber-rattling.

It is perfectly understandable why many F/OSS contributors would strive to avoid patent litigation. Choosing to avoid the costs and risks associated with patent litigation is rational behavior because those costs and risks are substantial.

4.2 Patent litigation is costly and potential individual liability is high

It is little wonder that few F/OSS contributors have taken Professor Jacobsen's path. Consider the costs of his suit against KAM. The attorneys and the judge in his case have written and filed more than 250 documents—motions, memorandums, orders, and so forth. Working through all of these documents takes time; intellectual property legal services typically cost more than $200 per hour. The cost of unsettled patent litigation can easily cost hundreds of thousands to millions of dollars, even when relatively minor damages are at stake[12]. These costs must be absorbed, either through money raised or through the grace of qualified lawyers willing to forgo more lucrative opportunities in order to provide discounted services.

In addition to the costs of litigation, F/OSS contributors face an uncertain, and potentially high, level of liability when they choose to develop and distribute potentially infringing code. Patent infringement damages are generally based on whether the patent holder has lost profits as a result of the infringement and what a reasonable royalty for the licensing the patent would have been[41]. Proving such damages is difficult, but once proven, a patent holder is entitled to judgment[7]. If the patent infringement was willful, courts may even increase the damages[7]. A F/OSS product's popularity would lend credence to a claim that damages are owed.

Many F/OSS contributors are involved as individuals, as opposed to agents of an organization. These contributors face direct exposure to patent infringement claims. Without backing by an organization that is willing to defend and indemnify them, individual F/OSS contributors are unlikely to face down infringement claims in most instances.

They cannot be blamed for their trepidation. F/OSS contributors can only be expected to expend time, money, and energy to the extent that they value the benefits they receive from sharing their efforts with others. Participants derive a very real benefit from the F/OSS gift culture[65], but F/OSS participants cannot be expected to be martyrs, deriving an infinite benefit from their generosity, allowing them to face formidable levels of risk. Moreover, F/OSS participants may find comparable bene-

fits by abandoning certain F/OSS projects—those efforts targeted by software patent holders—in favor of F/OSS projects with lower levels of risk—those efforts that have not attracted the attention of software patent holders.

Unfortunately, the aggregate effect of individual F/OSS participants avoiding risk could be damaging to the greater F/OSS community. An increase in aggression by software patent holders could drive F/OSS contributors away from key efforts, reducing both the utility of F/OSS and the number of F/OSS consumers. A reduction in the number of F/OSS consumers may reduce the benefit F/OSS contributors obtain from sharing their efforts with others. Thus, increased software patent holder aggression could discourage F/OSS contributors from engaging in any F/OSS efforts.

4.3 Software patent holders are becoming more aggressive

Regardless of how expensive patent litigation becomes or how risk-averse the F/OSS is, software patents would have little reason to cause concern if software patent holders were themselves passive. So far, most software patent holders have appeared reluctant to assert patent rights against F/OSS contributors. While the number of patent suits is generally on the rise, few cease and desist demands have been sent to F/OSS participants. The Chilling Effects database currently only has seventeen entries for patent cease and desist notices—a whimper compared with the 1680 copyright and 378 trademark notices documented by Chilling Effects[21].

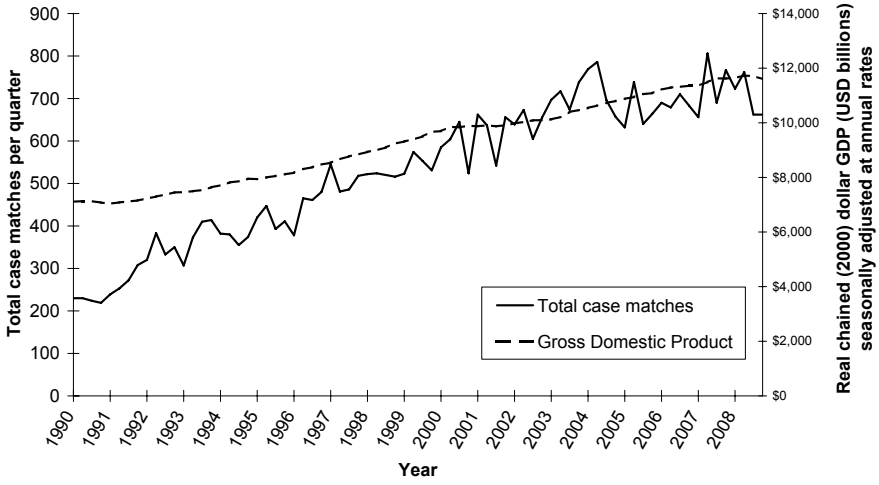

Fig. 1 United States Party/Case Index patent suit total case matches each quarter [18] and United States gross domestic product in billions of dollars, chained to 2000, seasonally adjusted at annual rates[66]. This figure shows that the number of patent cases appear to be growing at a slightly higher rate than the U.S. economy. No distinction has been made here between cases that involve software patents and cases that do not involve software patents.

There are, however, warning signs of increased aggression. Microsoft, which owns the rights to thousands of patents, has repeatedly claimed that the F/OSS Linux operating system infringes more than 200 Microsoft patents[48, 54]. This is a plausible claim given the role Microsoft has played in the formation of de facto software standards and its efforts to seek patents that cover such standards[51]. Red Hat, which markets a leading Red Hat software distribution, has found itself in patent suits multiple times[22, 25, 36]. Smaller players in the F/OSS community have also found themselves the targets of patent infringement claims[15, 40].

F/OSS participants are likely to encounter more frequent attacks, forcing individuals to either make a stand or abandon key efforts. Since F/OSS participants are risk-averse and face high risk in facing down such attacks, cracks in the foundations of F/OSS could develop under the strain of increased assaults. Safe harbor protection would strengthen the F/OSS foundations by shifting the risks of patent infringement to users who are actively interested in using potentially infringing technologies.

5 The F/OSS community should shift away from efforts to eliminate software patents entirely

Patent rights are extensive and powerful. They create risk which many F/OSS community members strive to avoid. If a significant number of F/OSS community members are discouraged from their efforts, then society will lose out on the benefits provided by free and open source software. A natural response to this situation is to oppose patent rights for software[50].

This is a draconian response that this paper does not endorse. The F/OSS community does not need "passage of a law to exclude software from the domain of patents[11]" to mitigate the effects of software patents on F/OSS. Eliminating software patents altogether may not even be in the best interests of the F/OSS community. It is better for F/OSS advocates to support reforms, such as F/OSS safe harbors, rather than oppose software patents on principle.

5.1 Mere arguments against software patents will not defeat them

F/OSS advocates have attacked software patents for decades[32]. Some argue questions of philosophy, such as the question of whether applied software and pure mathematics are distinguishable [43]. Some question whether the software industry needs software patents, a question that feels suspiciously similar to questions about the need for software copyrights[30]. Some attack the capabilities of the Patent and Trademark Office and the courts to deal with software innovations[11]. Others attack the motives of software patent proponents, alleging that they seek software patents "for the sake of profit, not innovation[61]." And there are some who have raised concrete concerns,

such as the possible impairment of F/OSS efforts or competition by producers of interoperable software[19, 50, 51].

The arguments raised against software patents have not been enough and may never be enough to convince policymakers to eliminate software patents altogether. For a critical mass of policymakers to support such a drastic shift, software patents would need to be clearly unworkable, the software industry as a whole would need to be in peril because of software patents, or software opponents would need to show that the potential gross benefits of software patents are negligible.

Software patents are not clearly unworkable. The patent system is under constant review by Congress, the Patent and Trademark Office, the courts, and other policymakers. This scrutiny produces changes to the patent system that mitigate many of the worst concerns. It has become easier to invalidate patents claiming obvious inventions using common sense reasoning[44]. It is more difficult for non-practicing patent holders to force infringers out of a market[24]. Those accused of infringing a patent may be able obtain a license and still go to court seeking to have the patent invalidated[49]. The Patent and Trademark Office is experimenting with ways of getting patents issued more quickly[9], which would make it easier to avoid potential infringement claims. Countless potential changes could continue to erode at arguments against software patents.

The software industry as a whole is not in peril because of software patents. It is true that patent litigation costs a lot of money, but this just makes software more costly and less profitable. Software companies fail all the time, but software patents are not the leading cause of failure[70]. And the biggest problems to hit the software industry recently, the burst dot com bubble and the broad economic downturn, were unrelated to software patents.

Finally, the potential gross benefits of software patents are not negligible. Look at Google, the rising star company that dramatically raised expectations of what a search engine should do. Google's success may be attributable in part to the fact that it is the exclusive licensee of the patent on the PageRank method of using hyperlinks to rank search results[77]. If Google had not had exclusive access to this technology, its competitors, who were already well established, might have been able to create their own ranking systems more quickly, driving Google into obscurity. For Google, software patents did not provide value through licensing, but through the ability of Google to differentiate itself from the competition.

Software patents may promote software reuse, reducing wasted reinvention energies in the software industry[29, 33, 63]. Software patents may even spur innovations simply by putting roadbloacks that have to be traversed. The inconvenience of dealing with intellectual property claims to the LZW compression algorithm spurred development of the PNG graphics format as an alternative to GIFs [51]. Browsers now widely support the PNG format, which uses a superior compression algorithm, as a result of software patents.

Circumstances may change, but the forecast for software patent abolition is bleak.

5.2 Absent software patents, patent-level protections would likely seep into other forms of intellectual property

It is just as well that software patent opponents are unlikely to completely exempt software innovations from the patent system. If software patents were eliminated completely, an intellectual property vacuum would be left behind. Such a vacuum might be replaced with a new system that is more suitable to the needs of the software industry and that allows for independent reinvention[35]. Unfortunately, a software patent vacuum might, instead, be filled by extending the scope of other forms of intellectual property protection, such as copyrights, trademarks, trade secrets, and contractual obligations non-compete agreements.

This is especially likely in the United States, where judges have broad powers to interpret law. Judges are undoubtedly influenced by their own sense of what is just[67]. If a law can be reasonably interpreted in a way that leads to a just outcome, then a judge may choose that interpretation over another.

What this means is that the power of Lockean logic—that one should own what one has created[46]—can be a powerful factor. The Lockean model of intellectual property rights may be criticized[31] or even leveraged to support positions critical of software patents, but the role that this model plays in expanding the interpreted boundaries of intellectual property rights should not be ignored.

Consider the expansions in copyright law that took place before software was widely considered patentable. These expansions began to encroach on the non-literal aspects of software, resulting "in protection beyond that provided by the patent law, without disclosure or examination, and for a much longer period of time[33]." The destruction of software patents would make copyright claims, such as those made by The Tetris Company, seem more reasonable to judges who might wonder why certain software innovations seem unprotectable by any measures.

Autodesk, an early opponent of software patents[64], recently presented another example of an effort to expand non-patent intellectual property to create patent-like protection. When Autodesk released AutoCAD 2007, Autodesk introduced TrustedDWG technology into the DWG file format. This technology makes it possible for Autodesk to warn users when a computer-aided design file was produced by a tool that did not license Autodesk's technology. It works by inserting "an identifying watermark and a proprietary string of code known as the TrustedDWG code" into the DWG file[13].

Soon after the release of AutoCAD 2007, the Open Design Alliance (ODA) announced that a new version of its DWGdirect library, which read and writes DWG files, also included TrustedDWG technology. The ODA position was that TrustedDWG technology had "nothing to do with determining if a DWG file is trustworthy, and serves only to unnecessarily alarm users, warning them that perfectly good DWG files 'may result in stability issues' when used with AutoCAD[13]."

The case settled in less than six months, with the ODA consenting to judgment that it had infringed on Autodesk's trademark rights and that it could no longer implement TrustedDWG technology[14]. Perhaps the ODA could have fought Autodesk's

infringement claim. After all, there was controlling case law that could have hurt Autodesk in court [69]. But the ODA chose not to fight, leaving Autodesk with patent-level protection that could potentially last forever.

Even if software patents in name go away, software patents in practice may live on.

5.3 The F/OSS community does not actually need to eliminate software patents

The patent system is, in many ways, like the income tax system. Instead of taxing income, the patent system taxes those who use innovations, whether they are alleged infringers who pay through litigation costs, actual infringers who pay damages, or consumers who pay monopoly prices. It taxes the generations of today to encourage technologies that, because patents have limited duration, will be free to use for the generations of tomorrow; a gift of technology from one generation to the next.[10].

Many believe the income tax system is flawed. It has attracted the attention of many outspoken critics, often with impressive credentials. Yet the chances of eliminating or drastically reforming the tax system are slim. That has not stopped various interest groups from seeking to protect themselves from the income tax system or at least mitigate its effects. They have been successful in securing exemptions—concessions that benefit them by shifting some of the burden of funding government activities to everyone else.

One potential reason why radical income tax reform efforts have gained little traction is the belief that, because of the political realities of interest groups pushing for a slightly different tax burden, a new tax system would ultimately begin to look like the existing tax system. This same consideration should be taken into account by those who advocate dramatic changes in patent law.

For example, consider that patent law is not just like a system of taxation, it is also like a system of providing prizes, in the form of monopoly profits, to those who create useful innovations first. The government can provide prizes directly, but doing so requires the government, not the people, to decide what innovations are useful enough to merit reward and what those rewards should be. Without patents, innovations created to engineer-around existing, protected technologies, also may not come into being. It is unclear that eliminating patents in an entire field, and justifying doing so with a prize system, provides the optimal policy. Concerns like these could make drastic changes to the patent system unpalatable to those trying to encourage innovation.

Instead of trying to eliminate software patents, F/OSS advocates should allow the software patent system to continue evolving, but push decision makers to keep F/OSS out of this evolution. This is where the proposed safe harbors fit in. They would not eliminate patent rights in software-based technologies, but would help reconcile the F/OSS model of software development and distribution with the commercially-driven models that the patent system is better suited for. Safeharbors would shift the costs of software patents to those who can best manage them.

6 F/OSS advocates should engage in principled negotiation

The proposed safe harbor protections for F/OSS are not necessarily the only types of reforms that F/OSS advocates should seek. F/OSS advocates should be creative and look to other ways that F/OSS interests can be protected. But, F/OSS should strive to always engage in principled negotiation when advocating patent reform.

Principled negotiation is the classic approach to resolving conflicts by looking for win-win solutions. This method consists of four steps[26]:

First, separate the people from the problem. Software patent opponents and proponents generally are rational people. They just have reached different conclusions on a controversial issue. F/OSS advocates should to be careful not to engage in *ad hominem* attacks when arguing for reforms. As tempting as it might be suggest that software patents exist because "the global patent industry seeks [them] for the sake of profit, not innovation[61]," such advocacy is counterproductive because it stifles meaningful discussion.

Second, focus on interests, not positions. "Software should not be patentable" is a position. And it is a position that is debatable. There are good arguments on both sides of this question, but focusing on whether software should be patentable diverts attention away from the question of how do software patents affect various groups, including the F/OSS community.

Third, invent options for mutual gains. Eliminating software patents altogether is an option, but it should not be the only option. Well-structured safe harbor protections are an option that benefits the F/OSS community without substantially diminishing the interests of other groups, such as patent holders. F/OSS advocates could make significant headway by suggesting reforms such as these—reforms that are less extreme and that would address the interests of both software patent proponents and the F/OSS community.

Fourth, insist on using objective criteria. F/OSS advocates should keep in mind that innovation is king when it comes to the patent system[51]. They should still insist on other important criteria too. Non-infringers should be able to freely practice competing innovations. The costs of fighting infringement claims should be incurred by those who can best fight them. Paper tiger threats of patent infringement should be easily dispatched.

Engaging in principled negotiation would enable F/OSS advocates to have a voice in reforming the patent system to protect the F/OSS community.

6.1 Now is the time for F/OSS advocates to adopt new approaches to patent reform

Safe harbor protections that protect the F/OSS community have a reasonable chance of being enacted if F/OSS advocates show support for them now. United States Congressional leaders have spent several years pushing for the adoption of patent system

reforms[55, 56, 57, 58, 59]. These leaders have been unsuccessful because of tensions between the biological/pharmaceutical (bio/pharm) industries and the information technology/electrical engineering (IT/EE) industries.

Bio/pharm companies wield individual patents of great power, using them to hold back generic drug manufactures long enough to recoup astronomical research and development costs. In the IT/EE industries, patents are collected like trading cards: occasionally a patent turns out to have great value; but usually value is seen in an organization's collection of patents as a whole. At the Conference on Patent Reform, held in 2005 at the National Academy of Sciences Building, bio/pharm patents were likened to gorillas while IT/EE patents were likened to fire ants. It is unsurprising that these two economic giants cannot agree on patent reform measures.

Yet, they must agree. In today's environment, patent reform must be acceptable to both EE/IT and bio/pharm[47]. Congress is somewhat limited by its international obligations, which require that patents "be available and patent rights enjoyable without discrimination as to ... the field of technology[73]." This means that the patent system must be substantially uniform, even though different industries may have different needs.

Fortunately, Congress "may provide limited exceptions to the exclusive rights conferred by a patent, provided that such exceptions do not unreasonably conflict with a normal exploitation of the patent and do not unreasonably prejudice the legitimate interests of the patent owner[74]." Properly designed safe harbor protections could become one of those limited exceptions.

F/OSS advocates may be able to secure safe harbor protections by showing support for patent reforms, provided that those reforms include safe harbor protections or some other provisions to protect the F/OSS community. The F/OSS break from the absolute stance against software patents would weaken the IT/EE position on patent reform. This would force breakthrough compromises, making way for patent reforms, including the F/OSS safe harbors.

7 Conclusion

F/OSS is a movement with both economic and social value. Despite the positive aspects of F/OSS, the movement is not invulnerable; software patents could be used to weaken the F/OSS community. The answer is not the total annihilation of software patents. Even if it were, policymakers do not appear poised to eliminate software patents altogether. If they did, other forms of intellectual property would probably expand to create comparable levels of protection. F/OSS advocates should thus focus on reform that targets F/OSS interests, not the anti-software patent position.

Patent reforms are in the works, gummed up primarily because of interindustry conflicts. F/OSS advocates can take advantage of this state of affairs to seek powerful concessions. The right kind of concessions, such as safe harbor protections for F/OSS activities, would enable the F/OSS movement to continue flourishing without fear of death by software patents.

Acknowledgements Many thanks go out to the faculty at the George Washington University Law School for providing several years of intellectual stimulation. My classmates have also earned my gratitude for the grace with which they put up with my unceasing barrage of questions and comments during class. While some odd question or comment might hav derailed an occasional class session, it seemed that we were always exploring and learning at great speed.

References

1. 17 U.S.C. §512(l)
2. 17 U.S.C. §§512(c), (g)
3. 35 U.S.C. §102
4. 35 U.S.C. §103
5. 35 U.S.C. §271
6. 35 U.S.C. §283
7. 35 U.S.C. §284
8. 35 U.S.C. §285
9. Accelerated examination. United States Patent and Trademark Office. http://www.uspto.gov/web/patents/accelerated/. Accessed 15 Nov 2008
10. Adelman M, Rader R, and Klancnik G (2008) Patent law in a nutshell. Thompson West. St. Paul, MN. 5
11. Against software patents (1991) The League for Programming Freedom. http://progfree.org/Patents/against-software-patents.html. Accessed 30 October 2008
12. AIPLA report of the economic survey (2007) American Intellectual Property Law Association
13. Autodesk, Inc. v. Open Design Alliance, No. 2:06-cv-01637, complaint (D. Wash. filed 13 Nov 2006)
14. Autodesk, Inc. v. Open Design Alliance, No. 2:06-cv-01637, stipulated motion and consent judgment (D. Wash. filed and entered 03 Apr 2007)
15. Barracuda networks defends free and open source software from patent threat by trend micro (2008) Barracuda Networks press release. http://www.barracudanetworks.com/ns/news_and_events/index.php?nid=246. Accessed 07 Nov 2008
16. Based on a count of distinct entries in the online service providers directory of designated agents. United States Copyright Office. http://www.copyright.gov/onlinesp/. Accessed 28 Oct 2008
17. Based on a search for the party name "Tetris" in the United States Party Case Index of the Public Access to Court Electronic Records (PACER) service. https://pacer.uspci.uscourts.gov. Accessed 30 Oct 2008
18. Based on a series of time-constrained searches for cases, where the nature of the suit was reported to be patent-related, in the United States Party Case Index of the Public Access to Court Electronic Records (PACER) service. https://pacer.uspci.uscourts.gov. Accessed 19 Feb 2009
19. Chapin M (2008) Sharing the interoperability ball on the software patent playground. 14 B.U. J. Sci. & Tech. L. 220. http://www.bu.edu/law/central/jd/organizations/journals/scitech/documents/Chapin.pdf. Accessed 14 Nov 2008
20. Chee B (2007) The business case for open source software. InfoWorld. http://weblog.infoworld.com/geeks/archives/2007/03/the_business_ca.html. Accessed 05 Nov 2008
21. Chilling Effects database of C&D notices. http://www.chillingeffects.org/search.cgi. Accessed 07 Nov 2008
22. Datatern, Inc. v. Bank of America, Corp., 5:08-cv-00070 (D. Tex. filed 21 Apr 2008)

23. Directive on electronic commerce. Directive 2000/31/EC of the European Parliament and of the Council of 8 Jun 2000. Art. 14
24. eBay, Inc. v. MercExchange, LLC, 547 U.S. 388 (2006)
25. FireStar Software, Inc. v. Red Hat, Inc., 2:06-cv-00258 (D. Tex. filed 26 Jun 2006)
26. Fisher R, Ury W (1991) Getting to yes: negotiating agreement without giving in, 2d ed. Penguin Books, New York
27. Frequently asked question. Open Design Alliance. http://www.opendwg.org/faq. Accessed 30 Oct 2008
28. GNU General Public License. Version 3. (2007) http://www.gnu.org/licenses/gpl-3.0.html. Accessed 25 Nov 2008
29. Guntersdorfer M and Kay D (2002) How software patents can support COTS component business. IEEE Software. May/Jun 2002, 78 – 83
30. Hersey J (1978) Dissent of Commissioner Hersey. Final report of the National Commission on New Technological uses of Copyrighted Works. 27 – 37. Available at http://digital-law-online.info/CONTU/PDF/index.html. Accessed 08 Feb 2009
31. Hintjens P (2007) In defense of software patents. Free Software Magazine. http://www.freesoftwaremagazine.com/columns/in_defense_of_software_patents. Accessed 14 Nov 2008
32. History of the League for Programming Freedom. http://progfree.org/History/history.html. Accessed 10 Nov 2008
33. Hollaar L (1996) Justice Douglas was right: the need for congressional action on software patents. AIPLA Quarterly Journal 24(1), 283–305. Available at http://digital-law-online.info/papers/lah/aipla-qj.html. Accessed 15 Nov 2008
34. Hollar L (2002) Legal protection of digital information. The Bureau of National Affairs, Inc., Washington, D.C. 167–168. Available at http://digital-law-online.info/lpdi1.0/treatise33.html. Accessed 20 Feb 2009
35. Hollaar L (2006) A new look at patent reform. 19 May 2006 version. Originally published in the Journal of the Patent and Trademark Office Society, Sep 2005. http://digital-law-online.info/papers/lah/mini-patent-2006-05-19.pdf. Accessed 15 Nov 2008
36. IP Innovation, LLC v. Red Hat Inc, 2:07-cv-00447 (D. Tex. filed 9 Oct 2007)
37. Jacobsen v. Katzer, 2008-1001 (Fed. Cir. 13 Aug 2008). http://www.cafc.uscourts.gov/opinions/08-1001.pdf
38. Jacobsen v. Katzer, 3:06-cv-01905 (D. Cal. filed 13 Mar 2006)
39. JMRI: A Java model railroad interface. http://jmri.sourceforge.net/. Accessed 07 Nov 2008
40. KAM claims patent to run over model railroads (2005) Chilling Effects. http://www.chillingeffects.org/patent/notice.cgi?NoticeID=2432. Accessed 07 Nov 2008
41. Keely M. Estimating damages in patent infringement cases: an economic perspective. http://www.cornerstone.com/pdf/practice_intellectual_property/CR_Est_Damages_Patent_Infringement.pdf. Accessed 10 Nov 2008
42. Klemens B (2005) New legal code. IEEE SPectrium Online. http://www.spectrum.ieee.org/aug05/1686 Accessed 14 Nov 2008
43. Klemens B (2005) Software patents don't compute. IEEE Spectrum Online. http://www.spectrum.ieee.org/jul05/1557. Accessed 14 Nov 2008
44. KSR Int'l Co. v. Teleflex, Inc., 550 U.S. 398 (2007)
45. Lea G (1999) How MS played the incompatibility card against DR-DOS: real bear-traps, and spurious errors. The Register. http://www.theregister.co.uk/1999/11/05/how_ms_played_the_incompatibility/. Accessed 05 Nov 2008
46. Locke J (1690) Second treatise on civil government
47. Maebius S. and Wegner H. (2008) U.S. patent reform: the 2009 landscape. Trans-Atlantic Comparative Patent Institute. 13 Nov 2008 draft

48. Masnick M (2004) Operating systems, FUD and patent protection. Techdirt. http://www.techdirt.com/articles/20041118/0930249.shtml. Accessed 07 Nov 2008
49. Medimmune, Inc. v. Genentech, Inc., 549 U.S. 118 (2007)
50. Morgan J (1999) Chaining open source software: the case against software patents. League for programming freedom. http://progfree.org/Patents/chaining-oss.html. Accessed 30 Oct 2008
51. Morgan J (2002) Open source software and software patents: finding the common ground in a patent pool. University of Utah Bachelor's Thesis. http://opensource.mit.edu/papers/morgan.pdf. Accessed 30 Oct 2008
52. Newbold M (2008) How to leverage open source in a for-profit business. Local keynote presentation. Utah Open Source Conference 2008
53. OpenDWG R13/R14/R2000/R2004/R2007 file format specification version 4.0. http://www.opendwg.org/files/guestdownloads/DwgFormatSpec13-2007.rtf. Accessed 30 Oct 2008
54. Parloff R (2007) Microsoft takes on the free world. Fortune. http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/. Accessed 07 Nov 2008
55. Patent Reform Act of 2005, H.R. 2795, 109th Cong., 1st Sess. (2005) (hearings held in Subcommittee on Courts, the Internet, and Intellectual Property)
56. Patent Reform Act of 2006, S. 3818, 109th Cong., 2d Sess. (2006) (referred to the Committee on the Judiciary)
57. Patent Reform Act of 2007, H.R. 1908, 110th Cong., 1st Sess. (2007) (passed in House, placed on Senate Legislative Calendar)
58. Patent Reform Act of 2007, S. 1145, 110th Cong., 1st Sess. (2007) (placed on enate Legislative Calendar)
59. Patent Reform Act of 2008, S. 3600, 110th Cong., 2d Sess. (2008) (referred to the Committee on the Judiciary)
60. Perens B (2005) The emerging economic paradigm of open source. http://perens.com/Articles/Economic.html. Accessed 05 Nov 2008
61. Principles of the Foundation for a Free Information Infrastructure. http://www.ffii.org/. Accessed 13 Nov 2008
62. Pub. L. No. 105-304, 112 Stat. 2860. (28 Oct 1998)
63. Public hearing on use of the patent system to protect software-related inventions: transcript of proceedings. Jan 1994. 30 – 31. http://www.uspto.gov/web/offices/com/hearings/software/sanjose/sjhrng.pdf. Accessed 15 Nov 2008
64. Public hearing on use of the patent system to protect software-related inventions: transcript of proceedings. Jan 1994. 47 – 50. http://www.uspto.gov/web/offices/com/hearings/software/sanjose/sjhrng.pdf. Accessed 15 Nov 2008
65. Raymond E (2000) The hacking milieu as gift culture. Homesteading the noosphere. Version 3.0. http://www.catb.org/~esr/writings/cathedral-bazaar/homesteading/ar01s06.html. Accessed 13 Nov 2008
66. Real gross domestic products, chained dollars [Billions of chained(2000) dollars] Seasonally adjusted at annual rates. National income and product accounts tables. Bureau of Economic Analysis. United States Department of Commerce. 1990 – 2008, annual series. Revision from 30 Jan 2009. Available at http://www.bea.gov/national/nipaweb/Index.asp. Accessed 19 Feb 2009
67. Scalia A and Garner B (2008) Making your case: the art of persuading judges. Thompson/West, pp. 26 – 30
68. Sega Enterprises Ltd. v. Accolade Inc., 977 F.2d 1510 (9th Cir. 1992)
69. Sony v. Accolade, 977 F.2d 1510, 1528 – 30 (9th Cir. 1992). Available at http://digital-law-online.info/cases/24PQ2D1561.htm. Accessed 15 Nov 2008
70. Ten reasons high-tech companies fail. High Tech Strategies. http://www.hightechstrategies.com/10_reasons.html. Accessed 15 Nov 2008

71. The Tetris saga. http://www.atarihq.com/tsr/special/tetrishist.html. Accessed 30 Oct 2008
72. Tiemann M (2008) $60B less for proprietary software = $60B more customer value. Open Source Initiative. http://www.opensource.org/node/364. Accessed 05 Nov 2008
73. TRIPS: Agreement on trade-related aspects of intellectual property rights. Art. 27
74. TRIPS: Agreement on trade-related aspects of intellectual property rights. Art. 30
75. U.S. Patent No. 4,558,302 (filed 20 Jun 1983)
76. U.S. Patent No. 4,814,746 (filed 11 Aug 1986 with priority date of 01 Jun 1983)
77. U.S. Patent No. 6,285,999 (filed 9 Jan 1998)
78. U.S. Patent No. 6,530,329 (filed 17 Apr 2002)
79. Unsafe harbors: abusive DMCA subpoenas and takedown demands. The Electronic Frontier Foundation. http://www.eff.org/files/20030926_unsafe_harbors.pdf. Accessed 26 Nov 2008
80. Why there are no GIF files on GNU web pages. GNU. http://www.gnu.org/philosophy/gif.html. Accessed 30 Oct 2008
81. Witherspoon N (2008) Over, for now. http://twofingerplay.blogspot.com/2008_08_01_archive.html. Accessed 30 Oct 2008