Open Invention Network:
A Defensive Patent Pool for Open Source Projects and Businesses

Deborah Nicholson

> " *I think it is important to realize that technology defined* "
> *as practice shows us the deep cultural link of technology,*
> *and it saves us from thinking that technology is the icing*
> *on the cake. Technology is part of the cake itself.*
>
> Ursula Franklin
> *The Real World of Technology*

This article explores how patents impact innovation within free/libre open source software (F/LOSS) businesses and projects. The number of software patent suits brought each year is increasing and is diverting millions of dollars in funds from developers to lawyers. With patent suits on the rise, the US Supreme Court has left the F/LOSS community in a position where it must either wait years for legislation or address the issue of patent suits itself. However, defending the Linux kernel and related technologies is a different challenge than the one that faces proprietary software businesses. This article describes Open Invention Network, an initiative that is designed to meet the particular challenges facing the F/LOSS community and businesses by providing a defensive patent pool.

Introduction

The threat of software patent suits impacts standards, dictates what software becomes part of GNU/Linux distributions, creates extra work, and makes the end-user's experience less than ideal, as will be shown in this article. In last month's issue of the *TIM Review*, Monica Goyal (2011; http://timreview.ca/article/503) thoroughly examined some of the legislative ideas being discussed with regards to patent reform. However, legislative change will take years to achieve. In the meantime, more software patent suits are brought about each year. F/LOSS companies are being sued by both proprietary competitors and non-practicing entities. The F/LOSS community needs a viable defense now.

In this article, we examine the role of software patents and their impact on open source projects and businesses. First, we focus on the general challenges related to software patents. Next, we examine the particular challenges software patents pose to open source pro-

jects and businesses. Finally, we discuss Open Invention Network (OIN), a defensive patent pool established to help Linux-based projects and businesses defeat or deflect the threat of litigation.

Software Patents in the United States

In 2010, the Supreme Court ruled on *Bilski v. Kappos* (http://tinyurl.com/7mo5vvx), a case considering whether a particular business method for hedging risk ought to be patentable. No case addressing the patentability of abstract ideas had been heard in twenty years. Many hoped to see the Court use this case to generally narrow the scope of what is patentable, and sixty-eight amicus briefs were filed in this landmark case. An amicus brief allows stakeholders can choose to act as a "friend of the court" and typically offers the stakeholder's perspective on how the court's decision on a particular case is likely to affect them. F/LOSS businesses and many others pleaded with the Court to use *Bilski v. Kappos* to restrict what is patentable to a "machine or transformation," or alternatively to hand down some new doctrine that

Open Invention Network: A Patent Pool for Open Source Projects and Businesses

Deborah Nicholson

would put software patents (and perhaps business-method patents) outside the scope of patentability.

An earlier case, often referred to as simply "State Street", had established the "useful, concrete and tangible" doctrine, which required that there be a practical application for an invention to be considered patentable (http://tinyurl.com/7a2o7br). Most business method patents and software patents were believed to be outside the scope of the older doctrine although the lower courts had not upheld that idea. See Box 1 for a brief history of US patent law. For further details, see Patent Absurdity (http://patentabsurdity.com).

Ultimately, the Court ruled that Bilski's method was not patentable. Moreover, the Court chose not to take any kind of stand on what ought to be patentable; the majority opinion states:

> "...*patent law faces a great challenge in striking the balance between protecting inventors and not granting monopolies over procedures that others would discover by independent, creative application of general principles. Nothing in this opinion should be read to take a position on where that balance ought to be struck.*"

It would be hard for the Court to more thoroughly express their desire to maintain the current scope of patentability. In the decision, Judge Stevens spoke about patents in the information age, "If a high enough bar is not set when considering patent applications of this sort, patent examiners and courts could be flooded with claims that would put a chill on creative endeavor and dynamic change." Not only will nothing be done about software patents, the Supreme Court does not believe that there is a problem. Thus, the US courts have struggled to find a way to help investors make good on their investments while still promoting competitive innovation in a way that keeps pace with evolving technologies.

Non-Practicing Entities and Other Patent Challenges

Non-practicing entities (NPEs) are businesses that do not ship software or hardware or develop any sort of technology. These companies buy patents taken out by other companies; they sometimes purchase patents for current technology and sometimes for old technology, preferably if those patents include vague wording that could apply to other contexts. Some NPEs acquire pat-

Box 1. A brief history of US patent law

1952	The Amendment to the Patent Act Legislation added the word "process" to the list of what is patentable. Previous patents had been limited to manufacture and composition of matter.
1972	*Gottschalk v. Benson* (http://tinyurl.com/7cev3cl) In 1972, the courts felt that algorithms should not be patentable, but this idea was slowly chiselled away over the next 38 years.
1978	*Parker v. Flook* (http://tinyurl.com/7nksmvy) Mathematical algorithms are patentable if the implementation is "novel and non-obvious". This suit was about whether or not having some kind of trigger signal when a catalytic converter is operating outside certain desirable parameters ought to be patentable. In the end, the algorithm was not deemed patentable but the *application* of it was.
1981	*Diamond v. Diehr* (http://tinyurl.com/73f2n5d) A computer program that controls a machine is patentable. This case was about software being used to control the process of curing rubber. Again, it is the *application* of the software in a novel way that makes this innovation patentable.
1982	The Court of Appeals for the Federal Circuit was set up to hear appeals based on subject matter, including patents. The upshot? Patents suits are largely presided over by patent attorneys and the case law that was created here gradually paved the way for the unfettered patentability of everything, including software.
1994	*In re Alappat* (http://tinyurl.com/75z3dvn) Installing software on a computer makes a new machine which is patentable. This is often derided as the Piano roll blues, from Judge Rich's observations that a player piano is the same device no matter what roll of music making paper is loaded on it.
1998	*State Street Bank* (http://tinyurl.com/7a2o7br) The useful, concrete, and tangible doctrine came from this case. This was an attempt to exclude business method patents from the realm of what is patentable. Both concrete and tangible had potential to also knock software out of the pool of what is eligible for patentability as well, but this case was not upheld.
2010	*Bilski v. Kappos* (http://tinyurl.com/7eep7cl) The "machine or transformation test" is not the only valid test for patentability. The bench decided that they would not narrow the scope of patentability at this time.

Open Invention Network: A Patent Pool for Open Source Projects and Businesses

Deborah Nicholson

ents in areas that they believe other companies may be moving towards. Seventy-five percent of the suits brought by NPEs are software suits.

Ostensibly, the NPE business model is to help individual inventors or very small firms to manage their "intellectual property." However, their main source of income is filing lawsuits. Just fourteen NPEs raked in a combined $7.6 billion from 2000 to 2010. That figure represents only 9% of what the companies who were sued actually lost; defendants in those suits lost an estimated $87.6 billion in litigation costs and lowered stock value (Bessen et al., 2011; http://tinyurl.com/7jtap6x).

In 2010, the number of companies in all realms (including software) that found themselves in litigation with an NPE increased by an average of 48% when compared to the average of the previous three years (PatentFreedom, 2011; http://tinyurl.com/8ax9n3f).

Litigation is expensive, and so many companies settle. This ensures that a poor-quality or out-of-date patent can continue to be used to sue other companies. Many times, an NPE will sue using the same patent again and again. Fighting these types of suits can help knock out bad patents, but the cost is high.

In addition to the challenges posed by NPEs, there are also suits brought by other software vendors hoping to squash, annoy, or perhaps assimilate their competition. For example, in 1997, Intel sued a microprocessor competitor called Cyrix (http://tinyurl.com/6tsv5wy). Four years of litigation later, Cyrix "won" the suit, but they missed the opportunity to make money on their innovation. Technology moves faster than lawsuits, and the time for that particular microprocessor had passed.

Patent lawsuits are costly, even for the winners. According to James Bessen and Michael Meurer in *Patent Failure* (2008; http://tinyurl.com/6m8zf7o), a lawsuit that does not go on for too long can "cost only one-half million to a million dollars" and a case that goes to trial can cost "several million dollars" while, "in extreme cases, legal costs can mount to tens of millions." Those figures are enough to start another company or sink an existing one. Companies that are being sued will often see their stock prices plummet, while also suffering indirect costs due to the distractions a lawsuit brings. Money and energy are being diverted to legal battles from software development, project management, sales, support, and community outreach. All these costs can make the difference between success and failure.

Does the money that is exchanged in lawsuits ultimately fund innovation at another company once the lawsuit is over? As found by Bessen and colleagues (2011; http://tinyurl.com/7jtap6x), the answer for lawsuits brought about by NPEs is no: "most of the private losses incurred by defendants in NPE litigation do not appear to be transfers to other parties." It is clear that patent suits are not good for the business being sued, but the more important question is whether or not they are good for the industry as whole or even more broadly for society. The evidence does not support the theory that NPE activity is good for business or for innovation. "While the lawsuits might increase incentives to acquire vague, over-reaching patents, they do not increase incentives for real innovation" (Bessen et al., 2011). Promoting innovation is the supposed goal of the US patent system.

F/LOSS Companies and Projects

Patent lawsuits are not challenges for proprietary software companies alone; F/LOSS projects and companies may also be targets for litigation. The entities that are the most tempting targets are those that generate substantial revenue, such as Red Hat (http://redhat.com) and Google (http://google.com). In many cases the success of smaller F/LOSS projects depends on upstream success; imagine the GNOME desktop environment (http://gnome.org) without a major operating system distribution, or imagine Android applications without the Android platform. Also, many large F/LOSS projects depend on a closely related volunteer community, which represents a considerable asset that cannot be converted into a legal department or liquidated to fight a lawsuit.

Smaller projects are less likely to be sued, but patent concerns are still often harmful and time-consuming. For example, the GIMP photo-editing project (http://gimp.org) no longer includes the image mosaic plug-in after its developer received a letter alleging patent infringement. Says Peter Kirchgessner who has developed a number of GIMP plug-ins: "It is not clear if the patent is applicable in this case. But I have neither the time, interest or money for legal action. So I complied with the cease and desist request." (http://tinyurl.com/2emqbz). Even without a letter, the huge legal fees associated with software patents suits creates a chilling effect in certain areas or can consume large amounts of volunteer time to avoid hot spots. In another example, the Wine project (http://winehq.org), which allows GNU/Linux users to run Windows applications, has

Open Invention Network: A Patent Pool for Open Source Projects and Businesses

Deborah Nicholson

been forced to eliminate a critical feature. "Concerns about the Borland patent have prevented developers on the project from adding structured exception handling (SEH) to the free software compiler." (http://tinyurl.com/7nlm2m3) Removing SEH leaves developers in the situation of depending on a non-free tool, a licensing problem for free software distributions or writing a costly work-around. Even an unproven or vague suggestion of patent infringement can create a significant amount of additional work for a small project.

Even when no lawsuit is brought, the threat of a suit can cause problems for F/LOSS projects and companies. The 2009 debate over video formats for the web is a prime example of how patents can negatively impact end users. Apple worried that the Theora video compression format may be patent encumbered, which effectively stopped the adoption of Ogg Theora as the official HTML5 video codec (http://tinyurl.com/p8kfce). The firm MPEG LA has implied that all video standards are likely to infringe on existing patents (http://tinyurl.com/24yjld4). MPEG LA licenses related patents, so it is in their interest to make others wary of potential infringement and encourage them to pay licensing fees to use the technology. Would close scrutiny reveal that the Ogg Theora codec infringes on existing patents? Until a suit goes to court, there is no way to be sure. Meanwhile, the potential for patent infringement prevents content creators from using a single format that can be processed by all major browsers and developers for projects such as Fedora, Blender, and Miro spend time carefully excluding certain types of video support that would benefit their users (http://tinyurl.com/nelhsk). Also, lawsuits brought against one project can create work for other projects, result in exclusions to their final products, and ultimately impact their competitiveness in the market.

In light of the tremendous money to be made from patent suits, one might think F/LOSS projects ought to just "play the game" and start suing other companies for patent infringement. However, many free software contributors consider patent aggression morally repugnant. A company or project that relies on community support would endure a lot of backlash if it were seen to be a patent aggressor, especially if its actions negatively impacted other F/LOSS projects. F/LOSS communities differ from proprietary software businesses in several important ways, the foremost being motivation. Developers may just be "scratching their own itch" (i.e., working to solve a problem they personally experience) or they may be working to provide the wider com-

munity with a solution that may not be met by proprietary software, regardless of the community's ability to pay. Ethical concerns over control and access to computing motivate many contributors. These various motivations lead to different project structures and business models, filling every point on the spectrum from reliance on unpaid community members to fully funded staff. Most software projects are a hybrid, with community members moving from one project to another; some community members may be paid, some may not be. Community goodwill is critical for success in the F/LOSS world and its culture makes a strategy based on patent aggression unworkable.

Furthermore, the reluctance to wield patents as a weapon is often contractual. Many free software licenses have addressed patent aggression in their terms. The latest version of the GNU General Publish License (GPLv3; http://gnu.org/licenses/gpl-3.0.html) forbids a company from making patent infringement claims related to code that it contributed to a project under that license. The GPLv3 also asserts that patent rights that are extended to one recipient of GPL code must be extended to all recipients of that code. The Apache License (http://apache.org/licenses/) and the Mozilla Public License (http://mozilla.org/MPL/) also include clauses discouraging use of code under their license being used as a basis for a patent infringement suit. Apache terminates your license when litigation is filed:

> *"If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed."* (http://www.apache.org/licenses/LICENSE-2.0)

Many F/LOSS projects have neither the willingness nor the legal leeway to recoup losses from patent infringement suits by bringing suits against other software projects. For the free software community, the rise of software patent suits is a nuisance, not an opportunity.

As annoying as software patent suits are for F/LOSS projects, free software does not present a higher risk for infringement compared with proprietary software. As Dan Ravicher (2004; http://tinyurl.com/87ltxfk) points out, free software is at the same risk, since patents cover the idea or function rather than copyright, which covers the actual lines of code. Proprietary software and free soft-

Open Invention Network: A Patent Pool for Open Source Projects and Businesses

Deborah Nicholson

ware perform many of the same functions so both types of projects are equally vulnerable to suits for infringement. The good news is that a significant amount of code is being used by many projects, including some with significant resources to protect. For example, the number of projects using Autoconf (http://tinyurl.com/2psee4), the X window system (http://tinyurl.com/4mt9y), or OpenGL (http://tinyurl.com/733y8mo) is vast.

When a suit is brought, the court can choose to issue what is called preliminary injunction, or a mandate to stop the activity that is objectionable to the prosecution before the case is heard in full. As Ravicher (2004) points out, a preliminary injunction against a particular program would be impossible to enforce and there is no way to obtain a meaningful estimate of how many copies of any given piece of code are out in the world. The defendant would be unable to comply with such an order. A permanent injunction can be handed down after a suit is decided. After such a decision, the community would need to code around that particular patent. It is far better for broad and vague patents to be overturned through effective defense and prior art (http://wikipedia.org/wiki/Prior_art). Prior art refers to anything that has been made available to the public regarding a particular invention, including anything that proves an invention is obvious or not novel. Prior art can keep bad patents from being issued, overturn recent wrongly issued patents, and help with a pending lawsuit.

For F/LOSS projects and companies, lawsuits consume vast amounts of money and time that could be better spent on development, promotion, documentation, or translation. Pamela Jones from Groklaw says: "Knocking a patent infringement case out depends on having the precise weapons to do so. You can't fight something with nothing. If they are going to aim patents at you, you can't just stand there and hope for the best." (http://tinyurl.com/3xj5brl).

Open Invention Network

To help F/LOSS companies and projects overcome the challenges of patent lawsuits in a way that is compatible with the culture of free software, Open Invention Network (OIN; http://www.openinventionnetwork.com) was launched in 2005. OIN is an intellectual property company that was formed to further software innovation and promote Linux by using patents to create a collaborative ecosystem. OIN established a defensive patent pool to help F/LOSS projects, particularly those associated with Linux. OIN does not seek revenue by asserting its patents, but rather its intent is to allow community members to use its patents in a defensive way against those who attack Linux. Patents owned by OIN are available royalty-free to any company, institution, or individual that agrees not to assert its patents against Linux and related technologies. This enables companies to make significant corporate and capital expenditure investments in Linux – helping to fuel economic growth. OIN is backed by investments from IBM, NEC, Novell, Philips, Red Hat, and Sony. These six companies decided it would be mutually beneficial if they agreed not to sue each other over Linux and related technologies.

An example of OIN's role comes from late February 2009, when Microsoft filed a patent infringement suit against TomTom on eight patents, including three related to File Allocation Table (FAT) technology. Microsoft simultaneously sought an US International Trade Commission injunction against TomTom shipping product into the United States. TomTom reached out to OIN, as well as Linux Foundation and Software Freedom Law Center, for assistance with the suit. On March 23, 2009, OIN publicly distributed a press release indicating that TomTom had joined the OIN community of licensees. Microsoft settled the suit against TomTom shortly thereafter. TomTom was not required to disclose the terms of its settlement with Microsoft because the terms were deemed to be "nonmaterial" based on disclosure requirements in the Netherlands. Many believe that this particular suit was brought just to scare Linux kernel users. Bruce Perens observed: "What Microsoft really wants from TomTom isn't money, it's support in building fear about Linux in other companies, especially the makers of mobile and wireless devices just like TomTom's own product." (http://tinyurl.com/cq8d7v). There is a struggle going on for what kind of software we will use in the future. Given that lawsuits are expensive, the courts represent a stacked deck for the wealthier litigant.

In another example, Red Hat and Novell were sued in 2007 by IP Innovations, an NPE that owns 536 patents (http://tinyurl.com/76svjho). OIN supported the search for prior art to help invalidate the three patents using Linux Defenders (http://linuxdefenders.org), an online clearing house for prior art. Post-issue prior art, a term referring to evidence garnered after a patent has been issued, was crowdsourced from the community. Three junk patents based on X windowing systems from 1987 were knocked out (http://tinyurl.com/2g9jumu). IP Innova-

Open Invention Network: A Patent Pool for Open Source Projects and Businesses

Deborah Nicholson

tions will not be able to sue anyone else over those specific patents, but there are still many more to be struck down. It is notable that IP Innovations is a subsidiary of Acacia Technologies; there has been some speculation about the relationship between Acacia and Microsoft (http://tinyurl.com/8x3vqxw), which could mean deep pockets in addition to many technology patents.

Given the interconnections between F/LOSS projects, OIN would like more projects to become licensees so the F/LOSS community can focus on the external threats as a united front. For F/LOSS companies and projects, this means that OIN's defensive patent pool may be licensed for free. It is in the interest of our founding companies to see suits against the F/LOSS community defended adequately. Future cases over the same patents may refer back to decisions made in previous suits. Precedents built by suits against companies unprepared to fight back hurt the whole community.

OIN is staffed by a small group of F/LOSS community members, attorneys, coders, and outreach personnel who support OIN while also participating in other segments of the community. As with many other examples of the F/LOSS community working together on shared goals, it is impossible to gauge how much mutual success each organization is responsible for. OIN's success cannot be quantified as a separate element from the overall community's continued success. Given the current environment, where patent aggression is on the rise, OIN is proud to play its role in mitigating the risk of patent aggression to the Linux system.

Conclusion

Patent aggression exacts a substantial toll. As calculated by Bessen and colleagues (2011; http://tinyurl.com/7jtap6x), defendants in lawsuits with NPEs lost an estimated $87.6 billion in litigation costs and lowered stock value. Consider the social utility of $87.6 billion worth of coders, designers, and builders. If those suits are being brought strategically to erode the resources of the F/LOSS community, then this is a fight for the viability of free software. If the courts are not motivated by this cause, then another way must be found, such as that offered by OIN: a defensive strategy for F/LOSS projects and companies.

About the Author

Deborah Nicholson works at the intersection of technology and social justice. She has been a free speech advocate, economic justice organizer, and civil liberties defender. After working in Massachusetts politics for fifteen years, she became involved in the free software movement. She is the Community Outreach Director at Open Invention Network and the Community Manager at Media Goblin. She also serves on the board at Open Hatch, a nonprofit dedicated to providing tools and education for potential new free software contributors. She lives in Cambridge, Massachusetts where she is slowly pursuing a graduate degree in Information Technology.