Software Patents: Current Challenges and Future Solutions

Monica Goyal

> *We petition the Obama administration to:*
>
> Direct the Patent Office to Cease Issuing Software Patents
>
> *The patent office's original interpretation of software as language and therefore patentable is much closer to reality and more productive for innovation than its current practice of issuing software patents with no understanding of the patents being issued.*
>
> *Under the patent office's current activity, patents have become a way to stifle innovation and prevent competition rather than supporting innovation and competitive markets. They've become a tool of antitrust employed by large companies against small ones.*
>
> *To return sanity to the software industry – one of the few industries still going strong in America – direct the patent office to cease issuing software patents and to void all previously issued software patents.*
>
> Signed by 14,862 US citizens
> http://tinyurl.com/3u72683

Software patents for years have been used in the software industry to suppress innovation, kill competition, and generate undeserved royalties. This article considers whether software patents maintain the right "bargain between the inventor and the public" where, in exchange for disclosure of the invention to the public, the inventor receives a limited monopoly and the exclusive right to exploit the invention. This article argues that they do not and then explores possible solutions to address the problems identified. Those solutions include streamlining the patent process, making it more difficult to patent software innovations, making it easier to invalidate software patents, and shortening the patent protection from 20 to 10 years. The article closes with a call to action for people to work collectively to effect change in the industry.

Introduction

We have lost sight of the true meaning and purpose of patents. Patents were created in order to encourage innovation, not kill it. They were meant to protect the inventor, not further strengthen those with power. Patents have instead been used for years now in the software industry as a blunt weapon to suppress innovation, kill competition, and generate undeserved royalties. It is time to revisit the value of patents as they relate to software and test some of the policy reasons for awarding patents in the software context.

A patent is a "bargain between the inventor and the public" (*Free World Trust v. Électro Santé Inc., 2000*; http://tinyurl.com/cjvksfj) where, in exchange for disclosure of the invention to the public, the inventor receives a limited monopoly and the exclusive right to exploit the invention. The patent is a way for the inventor of a new device or method to reveal that device or method to the public so that, through the sharing of new ideas, other inventors, businesses, researchers, and academics can make developments in their own fields. In exchange for disclosing the fine details of their invention, the inventor receives the right to stop others from making, using,

Software Patents: Current Challenges and Future Solutions

Monica Goyal

or selling that invention for 20 years. It is said that, without the possibility of patent protection, people would not take the risk of time and money to create new products. The rights granted under a patent are very powerful, and when viewed against our free trade, or free economy principles, the effects are said "to take away free-trade, which is the birthright of every subject" (*Free World Trust v. Électro Santé Inc.*, 2000).

A common criticism regarding software patents is that software is not meant to be patentable and is not an invention as defined in the *Patent Act* (http://wikipedia.org/wiki/Patent_Act_(Canada)). Other critics claim that identifying software components that are novel or not obvious is difficult. Others state that the investment of time and cost is too small to warrant the quid pro quo of the monopoly granted with a patent. Still others point to the royalty and legal costs and the escalating restraints on trade to argue against the patenting of software.

Despite the admirable policy reasons underlying the *Patent Act* and the desire to award inventors with protection, the act currently fall short of its goals. Further, the implementation of the system is susceptible to manipulation. In this paper, we will first consider the patentability of software, then the costs of patent protection, the importance given to software patents by inventors, and the limits and consequences of the patent system. We will then canvas solutions and discuss the strengths and weaknesses of those proposals.

The Patentability of Software

The primary technical objective of the patentability of software is whether it qualifies as an invention as defined in the *Patent Act*; that is, any new and useful improvement or "any new and useful art, process, machine, manufacture or composition of matter" (http://tinyurl.com/c3vh9fp). Not all innovations or inventions are accorded patent rights. For example, mathematic algorithms, scientific theorems, and designs are not patentable. The difficulty is that a software program can use complex systems to emulate what would be physical processes or a machine, and thus it can become difficult to determine whether to classify the software program as a new invention or an algorithm or a design. The machine-or-transform test articulated by the US Courts and confirmed in *In re Bilski* (http://tinyurl.com/bqvk5wj) asks whether the software is tied to a machine that is not trivial or not conventional, or whether the software transforms an article from one thing to another. This kind of test highlights the difficulty the courts have in trying to draw a line between software as a patentable invention versus software as a design or concept.

Patents are Expensive

To play the patent game, one needs to have money. The cost of filing patents is estimated at $5,000 to $15,000 (Quinn, 2011; http://tinyurl.com/c6bus3m), where software patents tend to cost closer to the higher end of the spectrum. The cost of patent litigation is estimated prior to a trial at $1 million, and for a full patent defence, $2.5 million (http://tinyurl.com/3wj69c6).

Often, inventors starting out have very little capital. For example, a startup with even $100,000 in seed money that then pays $10,000 to $15,000 for patent protection has to make extremely difficult financial tradeoffs to do so. Not surprisingly, a survey of 1332 early-stage technology companies found that only 24% of software startups filed a patent (Graham et al., 2009; http://tinyurl.com/m9x65h). The most vulnerable are unable to afford patent protection, let alone file for a patent in the first place.

Those startups that do patent will often dream up ways to decrease costs. As a result, they may only file a provisional patent or fail to conduct an exhaustive patent search. In the latter case cutting corners can have significant impact on the effectiveness or "strength" of the patent and its enforceability.

Enforceability is where the real problem lies. A patent is not worthwhile unless you can enforce it. The cost of litigation is staggering. The only companies that can afford to enforce patents are those with deep pockets, and that very rarely describes a software startup, even if backed by venture capital. Very few companies can afford to defend a patent, and, as a result, many businesses weigh the costs and decide to pay the royalties demanded, even for what may be an invalid patent. To make matters worse, even the whiff of patent infringement is enough to quash a merger, acquisition, or business venture, which provides further incentive to pay royalties.

Software Patents: Current Challenges and Future Solutions

Monica Goyal

Innovate First and Patent Last

A reason for awarding patents and the ensuing monopoly is that "without the possibility of patent protection, many people might not take the risk of investing the time or money necessary to create or perfect new products," as stated in the Canadian Intellectual Property Office's "A Guide to Patents" (http://tinyurl.com/bty9vn8). Patent protection is generally an afterthought to engineers or computer scientists in the software industry at small or large companies alike. Instead, rapid prototyping and being first to market are orders of magnitude more important. Furthermore, lack of patent protection does not impede companies from entering a market or competing in that market. For example, consider Facebook's 800 million users, 75% of which live outside of the United States (http://tinyurl.com/356y6s), with users in countries such as India, Turkey, and Brazil. Lack of patent protection has not impeded Facebook from operating and being successful in these countries. Another example can be seen in the mobile app space, where a developer can create an iPhone app that becomes available for download anywhere in the world through Apple's App Store. The lack of patent protection does not stop people from creating and publishing new apps. What these examples highlight is that other solutions, including other business models (as seen in the App Store example or with the freemium model), can be used as effective ways of maintaining a competitive edge, and they can be more effective than patent protection.

There Are More Losers Than Winners

Today, it seems to be common rhetoric that if you are successful, you will eventually be sued. If you have conducted business in this industry for any length of time, you likely know of a company that has become the target of a software patent suit. At times, the persons who come knocking on the door are those whose only business assets are patents - they do not actually make any products. They usually seek some form of royalty from a legitimate business enterprise. Intellectual Ventures, for example, is reported to own 35,000 patents and earned $700M in revenue in 2010 (http://tinyurl.com/3wj69c6). For companies like Intellectual Ventures, the business model is to acquire and protect (and perhaps even sell) patents rather than produce and try to sell the products themselves.

It is Not Just the Patent Troll

Let us consider the bargain again: the inventor receives a patent in exchange for disclosure, but if their patent is invalid (i.e., it does not teach anything that was not known beforehand), then the bargain fails. However, we have a patent system where the cost to invalidate a patent far exceeds the cost of the patent itself. It is no surprise, then, that big companies aggressively patent ideas, even for things incidentally related to their business. Table 1 ranks the top organizations that were granted the most US patents in 2010; the list reads like a who's who of the technology industry. The big companies are just as guilty of heavy-handed tactics, but are surprisingly also victims of the system. For example, in second quarter of 2011, Microsoft earned three times more from Android than from Windows Phone 7 (http://tinyurl.com/3wj69c6). Microsoft thus benefits more from enforcing their patent then from creating a competing product.

Table 1. Organizations with the most patents granted in 2010*

Rank	Organization	Patents
1	IBM	5866
2	Samsung	4518
3	Microsoft	3086
4	Canon	2551
5	Panasonic	2443
6	Toshiba	2212
7	Sony	2130
8	Intel	1652
9	LG	1488
10	Hewlett Packard	1480
11	Hitachi	1447
12	Seiko Epson	1438
13	Fujitsu	1276
14	General Electric	1222

*Data source: United States Patent and Trademark Office: Patenting by Organizations 2010 (http://tinyurl.com/7zp5tm6)

Software Patents: Current Challenges and Future Solutions

Monica Goyal

Solutions

This section examines four possible solutions to remedy the problems with software patents by analyzing the strengths and challenges of each.

1. Make it less expensive

Solution: Streamline the patent process to make the filing and enforcement process less expensive.

Assumptions: This solution assumes that one could devise a simpler, lower-cost filing and dispute resolution system. It also assumes that enough democratic interest could be generated to do so. Furthermore, it assumes that the changes made would not lead to a more cumbersome system than the one we currently have.

Strengths: Such a solution would benefit all patentees, even non-software patents, and it would address a pain point felt by all companies now.

Challenges: The primary impediment to the cost issue is legal fees. There are very few people who have the knowledge and expertise to be a patent agent, and as such they command high rates. Secondly, legislative change may fail to be comprehensive and the sad reality is that this type of change is susceptible to lobbying by those with special interests.

2. Make it harder to patent

In the US and Canada, there have been attempts by the Commissioner of Patents, and the Courts to restrict the number of software patents. Take for example Amazon's "one-click" ordering system patent (http://wikipedia.org/wiki/1-Click#Patent), which was the subject of a patent infringement lawsuit in 1999. Amazon was responding to an "Express Lane" shopping checkout feature implemented by Barnes & Noble and which featured a one-click ordering method. Many programmers cite this as an example of what is wrong with the patent system. On the surface, it seems like an obvious feature to programmers and thus not deserving of a patent. The Commissioner of Patents agreed and the patent was denied (although through a successful appeal to the Federal Court the patent application was sent for a second review).

Solution: Award fewer software patents.

Assumptions: This solution assumes that there are qualified people with the right expertise to make the right decision, or else that there is a set of strictly defined parameters that can be set to aid in the decision-making process.

Strengths: This solution would reduce the number of software patents without taking the potentially untenable position to deny all patent applications.

Challenges: It is not clear that the requisite expertise exists to execute this solution. There seems to be difficulty in establishing consensus between the Commissioner of Patents, the Courts, and Legislatures, as evident by the recent Amazon decision in Canada and the *Bilski* decision in the United States, as described earlier.

3. Make it easier to invalidate patents

Every computer engineer or programmer in the industry has had at some point in their career a moment where they sit back in disbelief that someone somewhere thought to patent something obvious and certainly not novel. To be fair, this may be more a case of clever lawyering than a deficiency with the patent office. Regardless, when someone can play a system to his or her own advantage, that system loses credibility. And once a patent is awarded, it is difficult to invalidate. There was a recent US Supreme Court opinion where Microsoft (with Google and Apple) argued for patent invalidity to be proven through a preponderance of evidence (http://tinyurl.com/748hfp4). What the case does speak to is the "you got a patent for *what!*" effect that even the likes of Microsoft, Google, and Apple are not immune to.

Solution: Make it easier to invalidate patents.

Assumptions: There are many invalid patents awarded, or we can easily assess the invalidity of a patent.

Strengths: This solution would discourage people from filing invalid patents.

Challenges: There is a danger that legitimate patents will be invalidated, especially by those with the financial means to seriously challenge an otherwise valid patent.

4. Decrease patent lifetimes

The length of the monopoly is no longer sustainable in light of the rate of development. Twenty years in the software industry is two lifetimes, maybe three. Fifteen years ago we still listened to music on cassette tapes. It

Software Patents: Current Challenges and Future Solutions

Monica Goyal

was not long before CDs became the standard, then MP3 players, and then downloading music took over. Now we stream music through online services such as Spotify and Pandora. It is clear that 20 years is a long time, and the commercial lifespan of software could be as short as five years. The result is that we allow companies to have a complete monopoly over multiple lifetimes of a device.

Solution: Decrease the lifetime a patent is awarded from 20 years to 5 or 10 years.

Assumptions: Less time is needed to recover development costs.

Strengths: This solution reduces the restraints of trade and the incentives for patent trolls. It also strikes a different balance between the inventor and the public in an industry where the research and development costs may be lower, and where there are concerns over awarding invalid patents.

Challenges: This solution does not address the patentability of software issue or the costs issue related to patents.

Conclusion

The Canadian Patent system is justified by the idea that it promotes research and development and protects an invention. The assumption is that without the quid pro quo of patenting, inventors would not take on the risk of inventing. Instead what we see is that, regardless of patent protection, companies will still create and innovate software products, treating patenting as an afterthought. Those who are most vulnerable actually go without patent protection, and very few can afford the high costs of patent enforcement. In general, the cost of patents is staggering and essentially diverts resources from productive enterprises. We can no longer claim that the Canadian patent system is designed to benefit Canadians. It appears to be only useful to the handful of companies who can afford it. We are crippling innovation in the software industry with our own rules and reducing our competitiveness at a global level. We will need a multi-pronged approach to address reform as it pertains to software and it will have to be a collectively organized effort in order to thwart special interest groups. Because right now the status quo does not serve anyone well.

About the Author

Monica Goyal is a Toronto-based lawyer and a softwarenology entrepreneur who founded My Legal Briefcase (https://www.mylegalbriefcase.com), an online legal service firm focused on small claims court cases. After graduating from her undergraduate degree from the University of Waterloo, where she was a Dean's List Scholar, Monica attended Stanford University, where she earned her Master's degree in Electrical Engineering. Monica also holds a law degree from the University of Toronto. Monica's volunteer work with organizations such as Griffin Centre, Adventure Place, Downtown Legal Services, and Pro Bono Law Canada has given her insight into the accessibility and affordability of legal needs for the marginalized. She developed My Legal Briefcase to empower individuals going to Small Claims Court.

Citation: Goyal, M. 2011. Software Patents: Current Challenges and Future Solutions. *Technology Innovation Management Review*. December 2011: 18-22.