Do patents stifle or encourage innovation in software?

Chris Kelly

PUBP 4410
Science, Technology, and Public Policy
December 5, 2005

Background

Patents are designed to encourage innovation but recently there has been a lot of debate surrounding software patents and their effect on innovation. Independent software developers are claiming that patents prevent them from innovating while large companies keep relatively quiet and continue to apply for patents by the thousands. The goal of this paper is to explain the background of the situation, show why it matters, compare both sides of the debate, and decide which side seems to have the most merit.

As scientific research and development continues at a seemingly unstoppable pace, our society is further progressing from a mechanical age to an information age. As explained by Moore's law, the complexity of integrated circuits (the basic building blocks of digital technology) is doubling ever 24 months while the cost of the same components is split in half.[21] With this exponentially increasing amount of computing power, computers are showing up everywhere: cell phones, cameras, music players, network devices, sensors, and many others. Each one of these devices requires software to run and communicate with other devices, and the number of different pieces of software that could be written for each device is almost unlimited. In 1999, the sales of PC Application software in the U.S. alone was almost $20 billion.[45] Combined revenues of software companies mentioned later on in this paper total over $250 billion a year. Essentially, software is a very large business and will continue to grow as technology becomes more and more pervasive. Innovation in software is necessary: new software needs to be written to meet new needs, power new devices, and allow new technology to interpolate with older technology.

From the U.S. Constitution Article I Section 9 Clause 8, the goal of the U.S. patent

system is “To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.” It would seem that in innovation driven industry such as software, there would be a large reliance on patents to protect innovations and prevent people from making money off of the innovation of others. However software is so different in many ways from other patentable material that a growing number of developers, companies, and experts feel that software should not be patentable at all.

Comparing Software Patents To Other Patents

For several reasons, it does not make sense to group software patents with most other kinds of patents. Firstly, as far as most non-software patents go there is a clear difference between describing the production of a product and actually producing it. However, a description of a software algorithm that meets disclosure requirements is indistinguishable from the actual implementation of the algorithm. Patents are supposed to be issued on inventions and methods, not “ideas.” Software exists only as information, a space usually limited to ideas which are protected by copyrights not patents. The speed of software research and development is also much faster than physical products and processes. This manifests itself as several additional differences in software patents from other types of patents including the probability of independent invention and the building block nature of software.

Given the nature of software and the standard sets of tools used to solve similar problems, independent creation of identical or very similar solutions is almost guaranteed. A programmer on Microsoft Windows using the Visual Basic language trying to find the biggest number in a list for a computer game high score screen could come up with the following solution:

```
Dim max as Integer = Array(0);
For position As int  = 0 To Arrray.Length() Step position
        If Array(position) > max
                max = Array(position);
        End If
Next position
```

While a programmer on Linux using the Java language trying to find the biggest number in a list for a log file analyzer could come up with:

```
int max = Array.itemAt(0);
for(int position = 0; position < Array.length(); position++) {
        if (Array.itemAt(position) > max) {
                max = Array.itemAt(position);
        }
}
```

These two code snippets are for two completely different solutions in two different implementations, yet their method is identical and this method of finding the largest number in a list could be patented by another developer and both of these developers would be liable. This actual example is not necessarily a patentable concept, but slightly more complex concepts such as a method of conveying electronic information from a device to a mobile telephone network [U.S. Patent 6,810,234] and an “on-line directory service through the Internet” [U.S. Patent 6,208,998] are patented every day. An example recently in the news is Amazon's patent on ordering items in an on line shopping cart with one click [U.S. Parent 5,960,411]. Each of these concepts is one that someone knowledgeable in a certain aspect of technology could come up with independently.

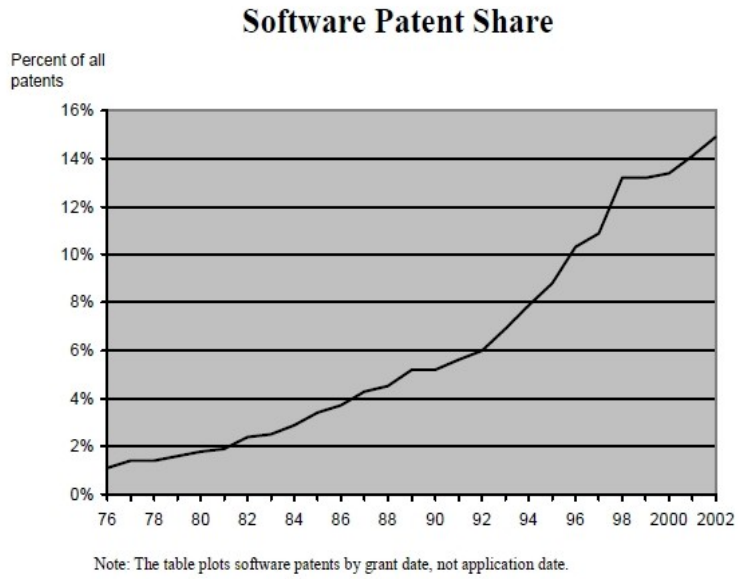
Another similar aspect of each of these examples is that each concept is part of a larger system. Software is made up of hundreds if not thousands of different parts, many of which can only be implemented in a certain way. An “Algorithm to find the shortest path between two

points" could be required for many different applications, but someone holding the patent on the algorithm could prevent further research on these other applications. For other types of patents, this may be acceptable. After 20 years, the patent would expire and everyone else would get to use the information. However, due to the rapid speed of software development, as little as a year could be enough to put an unfair distance between the patent holder and independent inventors of the same concept. Also, given how many software patents are currently being issued and the complex technical language used in the patents, there is no way for a software developer to stay up to date with all of them. A software developer may use a concept that he invented independently based on his own knowledge even though a patent was issued for it 2 years prior, yet this patent could prevent him from using what he came up with. Lastly, companies often push for patented technologies to become national and international standards. Almost all audio, video, and wireless protocols originated as the research of one or a few institutions or companies, many of which chose to patent their technology. Once these technologies become standards, anyone wishing to implement something to the standard will be required to license the technology from the patent holder first.

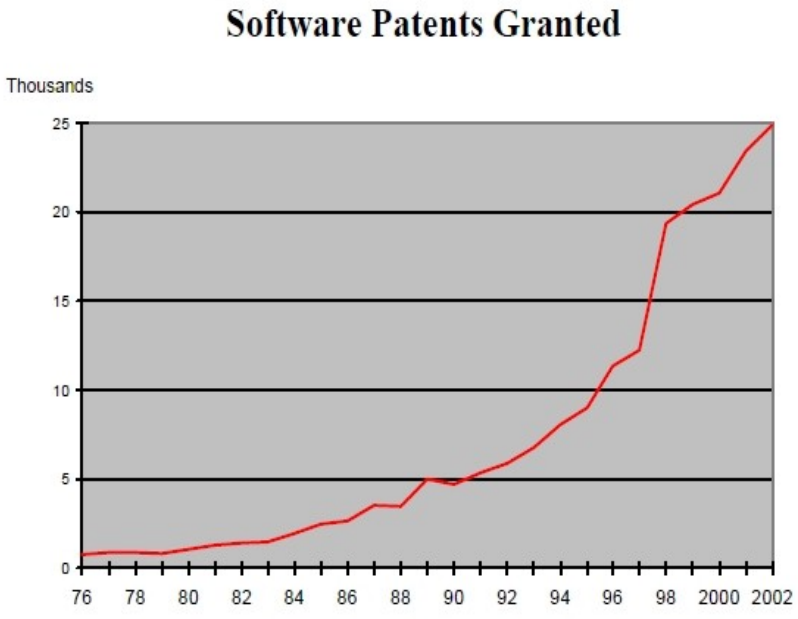
These reasons alone could be used to argue against software patents, however there are several reasons that software patents are a good idea. Software, like all other forms of innovation, is a result of time and money spent on research and development. A company or individual is not necessarily going to spend a lot of time and money to come up with an innovative solution to a problem only to give away the results. Innovation is innovation regardless of the final product and there should not be a penalty for choosing to innovate in a particular field. Information technology is quickly growing to be one of the most important innovation based industries and

the protection of ideas is needed to allow corporations to keep spending billions on research.

No matter which side is taken, both sides are backed by lots of information and diverse groups of supporters. The number of software patents being issued continues to increase dramatically as shown by these figures of patents issued in software and that number compared to the total number of patents: [46]



Note: The table plots software patents by grant date, not application date.



Note: The table plots software patents by grant date, not application date.

With this continual increase in patented software and the growing number of people taking up each site of the debate, this issue needs addressing soon.

Proponents and Opponents

As early as the time of Thomas Jefferson, people have been debating the patentability of ideas. In a letter he wrote to Isaac McPherson Monticello on August 1, 1813, he wrote:

> [...] If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possesses the whole of it. He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me. That ideas should freely spread from one to another over the globe, for the moral and mutual instruction of man, and improvement of his condition, seems to have been peculiarly and benevolently designed by nature, when she made them, like fire, expansible over all space, without lessening their density in any point, and like the air in which we breathe, move, and have our physical being, incapable of confinement or exclusive appropriation. Inventions then cannot, in nature, be a subject of property. [...] [19]

Software is exactly the “idea” the Jefferson describes, someone taking software and using it doesn't take away from the inventor, and the spread of this kind of knowledge should be encouraged. This conclusion that software is just an idea is accepted by most people opposed to software patents, and arguments similar to the one above are used to say that because someone can use an idea without direct harm to the originator of the idea, it should not be patentable. However, companies that spend millions of dollars on research into software disagree because they lose possible profits and recovery of research money if all the ideas they create are given away at no charge. In an article in June 1992 issue of Communications of the ACM, the author explained a similar situation he was in:

> I was then faced with having invested six years of raising money, developing a product, marketing it, and proving its value in the market, only to find I was in debt, my customer base was on a dying computer and Apple was giving away free a more polished and featured, although less elegant, version of the metaphor. [5]

His invention was patented, and he sued Apple leading to a settlement, but to put an interesting twist on it, he also sued IBM for a similar product of theirs. IBM is later in this paper shown as a supporter of software patents, but apparently only when it is in their favor:

> I spent six months patiently trying to deal with IBM. Finally IBM representatives flew to San Francisco to show us prior art-earlier technology-invalidating our patents that they claimed to have. When they arrived, they refused to show us the prior art, "for fear the patent office would recertify our patents in error." Even if IBM had been straightforward with me during the six months, to accept such an assertion without evidence would have been naive. [5]

It is likely that IBM was willfully infringing on the authors patent and unwilling to come to any agreement about it.

Even with his less than perfect experiences, the author came to the following conclusions mostly in support of software patents:

- Software patents stimulate companies to bring commercial products to market.
- Software patents stimulate new business formation.
- Software patents stimulate the commercial introduction of fundamental advances by small entities.
- Licenses are usually available where companies enforce patents.
- Small entities incurred little if any royalty and litigation costs for infringing patents.
- The patent system seems to reject bad patents early in the patent assertion process.
- If software patents were more widely respected we would probably have fewer variations on a theme, and more themes on which to vary.
- Big companies' patents do not seem to inhibit small developers.[5]

Each of these claims shows that software patents encourage innovation. They are backed up fairly well by the authors references and his article should be read for a more detailed explanation of his claims.

Some opponents of software patents (Oracle for example. [29]) suggest that copyright and trade secret protections are adequate protection for software. In a November 23, 2004 appeal the the EU Council, Linus Torvalds, Michael Widenius, and Rasmus Lerdorf wrote:

> At first sight, a patent appears to protect an inventor but the actual implications may be the opposite, dependent upon the field. Copyright serves software authors while patents potentially deprive them of their own independent creations. Copyright is fair because it is equally available to all. A software patent regime would establish the law of the strong, and ultimately create more injustice than justice. [2]

(Most individuals mentioned by name in this paper are or were an important player in the software industry and background information on each one is given in Appendix I.) However, this has two flaws: The goal of the patent system is to encourage innovation and this is achieved by granting protection in exchange for full disclosure of the idea so that others may use it later. Not many programming "ideas" could be completely protected by copyright as something can usually be rewritten in a different way to achieve the same result. Jefferson would like this because the creativity of the programmer would be protected by copyright but the idea would become public, but if developers chose to keep their code secret instead of acquiring the limited protection of copyrights, the rest of the world would never know about the ideas and innovation would suffer. Additionally, in some circumstances the secret code could be reverse engineered from the final product, putting the trade secret into the public domain or the hands of a competitor at which point there would be nothing that the creator could do to further protect their idea.

Other opponents of software patents take a different approach, not claiming that software should not be patentable, but that existing software patent systems harm small businesses and individuals that are not as well funded as an entity like IBM. In the same address the the EU

council, Torvalds, Widenius, and Lerdorf wrote:

> In particular, we believe that the economic opportunities of the new EU member states are endangered by software patents. The many talented software developers in those countries should be given a fair chance. The average cost of a European patent is in the range from [$35,106 to $58,510], and a company needs a very large number of such patents in order to be able to enter into "cross-licensing" agreements with multinationals that own tens of thousands of patents each. [2]

Due to the number of patented technologies required for some pieces of software, having a large portfolio for negotiation is required for two companies to come to an agreement and for either company to be able to make a product. Not only is the cost high, but software and the Internet have enabled small businesses, non-profits, and individuals to make software a part of their every day business where before it was limited to large companies. With this increasing usage, large companies that previously only targeted large competing companies, typically those with the money to defend against illegitimate patent threats, are now going after smaller entities. Then,

> Faced with million-dollar legal demands, [these entities] have no choice but to capitulate and pay license fees – fees that often fund more threat letters and lawsuits. And because these patents have become cheaper and easier to obtain, the patentee's costs can be spread out quickly amongst the many new defendants. Our patent system has historically relied on the resources of major corporate players to defeat bad patents; now it leaves these new defendants with few if any options to defend themselves. [32]

Many other Open Source Software companies (See company policy section below) and developers feel the same way. They are giving away the ideas they create that are built on existing ideas and feel that this increases the common good and raise the question: "Should a patent holder be able to sue someone that is not making any profit off of their patent?" Patent holding companies maintain that they are suing for lost profits, and Open Source Software companies are even beginning to spend money from non-innovation based revenue to acquire patents to protect their innovation in products that they give away for free.

The Free Software Foundation Europe put together a list of “bad things” about software patents that sums up most of the points above. They stated that software patents:

- establish monopolies on abstract ideas.
- prevent innovation by allowing grants without implementation (Source Code) and thereby preventing such research.
- prevent competitive markets by giving large players absolute control over the marketspace.
- prevent disclosure of ideas, the original motivation for the introduction of patents.
- prevent interoperability, increasing your dependence on a single vendor. [48]

Each of these points points to software patents stifling innovation and many activist organizations in Europe and the United States have independently come to similar conclusions.

Lastly, The League for Programming Freedom's submission to the Patent Office on January 25, 1994 pointed out a very interesting discrepancy: The companies with the largest numbers of software patents were not typically the companies that were driving innovation in software. The specifically wrote:

> None of the hardware or software companies that collectively constituted the microcomputer revolution hold significant numbers of software patents. Companies such as Microsoft, Borland, Novell, Adobe, Lotus, NeXT, Intel, Apple, Sun, and SGI all have relatively weak software patent portfolios. These are the companies that have created wealth in the computer industry over the last ten years by developing new and innovative products. They are very much responsible for turning the industry into the vibrant place it is today. Without these companies, the software industry would be virtually nonexistent. [24]

Things may have changed some in the 10 years that followed, but it is very hard to get accurate counts of software patents and companies that hold them due to the lack of adequate categorization in the data that the United States Patent and Trademark Office makes available to the public.

Company Policy Towards Software Patents

Almost all large software companies have some sort of policy towards software patents. None are specifically for the existing patent system and how it applies to software, but there is a varying degree of dissidence from suggesting small changes to completely abolishing software patents. At a public hearing on the use of the patent system to protect software related inventions, several of the major players in the field expressed their views. Adobe, Autodesk, Borland, Oracle, Synopsis, and Wind River Systems were all opposed to software patents; Sun and Time Warner were concerned with software patents; And IBM, Intel, Microsoft, SGI, and Taligent were in favor of software patents. [4] (Every company mentioned in this paper is or was a major player in the software industry and background information on each one is given in Appendix II.)

Those in favor of software patents all mentioned things like wanting "the issuance of poorly-examined patents curtailed," feeling like "the industry would benefit from a reduction in the average [review time] of applications before the Patent Office," and that various other changes in the patent office would ensure more effective examination of software patents so only ones that truly should be issued get issued. [14][20][4] A Microsoft press release from August 2005 presents the four points that most software patent supporters seem to have agreed upon:

> First, we want to ensure high patent quality, in part by making certain that the Patent Office has the resources it needs to examine increasingly complex patent applications in the technology space. Second, we support efforts to curb excessive litigation and litigation abuses that have emerged in the patent space. Third, we want to see greater harmonization of the various patent systems around the world. And finally, we support legislation to increase access to the patent system for small inventors. [23]

"Concerned" companies felt that the above mentioned points were enough to warrant concern, while opposing companies very strongly oppose software patents with statements such as "[Oracle] believes that existing copyright law and available trade secret protections, as opposed

to patent law, are better suited to protecting computer software developments," [29] and a representative speaking for Adobe feels that

> The software marketplace requires constant innovation regardless of whether the computer programs can be patented or not. Indeed, the fundamental computer programs and concepts on which the entire industry is based were conceived in an era when software was considered to be unpatentable. [...] I believe that software per se should not be allowed patent protection. I take this position as the creator of software and as the beneficiary of the rewards that innovative software can bring in the marketplace. I do not take this position because I or my company are eager to steal the ideas of others in our industry. Adobe has built its business by creating new markets with new software. We take this position because it is the best policy for maintaining a healthy software industry, where innovation can prosper. [1]

Red Hat has "consistently taken the position that software patents generally impede innovation in software development and that software patents are inconsistent with open source/free software" [40] and at a MySQL User's Conference 2005 event, one of the Vice Presidents of Red Hat said specifically that software patents stifle innovation and that "Every time a software patent blooms, it's a promise to cease innovation in that space for 20 years." [39]

Yet as strong as the opponents of software patents present their case, important supporters continue to stand behind their beliefs. IBM maintains that "The long term value of R&D in the marketplace is in the new functions implemented by software. If such new functions are protected, investment flows to the industry. If not, investment will dry up." [14] but an IBM representative also brings up the issue of separating generally accepted hardware patents with controversial software patents:

> There are several other points I want to make on this issue. We can't divorce computer program-related inventions from computer hardware and other microprocessor inventions. The overlap between the two is so great that cutting back on one automatically cuts back on the other. An alteration in the standards for patent eligibility will also put the courts [...] in a quandary. That is because computer program-related inventions can be implemented in either hardware or software. Applicants will simply cast their patent claims in terms of electrical circuitry. And if

> you limit claim coverage over computer program implementations of circuit inventions, you will turn electrical patents into nullities, because the circuit functions can be implemented by a programed computer. [14]

Intel supports existing patent laws with regards to software [16] and Microsoft agrees with them, stating that they "do not believe that patent protection should be withheld from an invention that otherwise meets the statutory requirements for patentability, simply on the basis that the invention is or may be embodied in software."[20] Microsoft also feels that the existing system has a long history which shows that it maintains an appropriate balance in "protecting inventive technology." [20] SGI feels that software patents are a necessity and states that the only reason there are problems with software patents is because of the number of "overly broad" software patents that have been issued. Given that software is built of many small pieces in the same way that hardware is, and that patents have been issued on transistors and other building blocks of electronics without serious impediment to the electronics industry, SGI feels that the same concept applies to software and that software does not need special consideration as a result of being built by small pieces. [43]

Software Patents And Open Source

One specifically interesting facet of the debate is the effect of patents on Open Source software, "software whose source code is available as open source under an open source license so that anyone can study, change, and improve its design." [28] There has been a large amount of concern in the Open Source world about patents from players small and large, and the Open Source world is probably where initial concerns about the patentability of software arose. Red Hat mentioned the incompatibility between patents and Open Source software, and other large companies have been acknowledging this problem by "pledging" their patents to Open Source:

promising not use patents in their portfolios against Open Source developers. In January of 2005, IBM (a supporter of software patents) pledged 500 U.S. patents to "any individual, community, or company working on or using software that meets the Open Source Initiative (OSI) definition of open source software now or in the future." [13] Then, in November of 2005, Novell, Philips Electronics, Sony and Red Hat got together to start the "Open Invention Network" as "a company that has and will acquire patents and offer them royalty-free to promote Linux and spur innovation globally." Instead of just offering these patents for people work on Linux, these patents are only available to "any company, institution or individual that agrees not to assert its patents against the Linux operating system or certain Linux-related applications." [15][26]

This all seems like a great idea to help Open Source software and the associated innovation, yet so far the patents pledged "represent only a small fraction of the estimated 150,000 to 300,000 software patents that have been issued in the U.S." [54] Patents that companies are keeping private could later be used as a patent "weapon of mass destruction," and Mitch Kapor points this out as problematic to the Open Source model:

> If totally pushed to the wall--because their business model no longer holds up in an era in which open-source is an economically superior way to produce software, and the customers understand it, and it's cheaper and more robust, and you've got the last monopolist standing--of course they're going to unleash the WMDs. How can they not? [27]

Like Kapor, Open Source Software developers tend the most vocal opponents of software patents. Some tell apocalyptic stories of the end of our basic rights, and others just feel that the system needs to be reworked. Typically, companies have almost ignored these claims because Open Source has been too "academic," underground, and minuscule in the software marketplace. However, as the popularity of Linux and related software continues to grow, these developers

will start to play a much more powerful role in influencing software patent policy.

Current And Pending Legislation

Regardless of the opinions of developers and software companies, there are laws in place that establish what can be patented. The primary goal of patents is to encourage innovation and by the United States allowing software to be patented, the government is suggesting that software patents encourage innovation. Originally, software patents were not allowed in the U.S. because the Patent and Trademark Office (PTO) felt that patents could be granted to processes but not scientific truths or mathematical expressions of it, and software was considered to be a mathematical expression. The U.S. Supreme Court upheld this in 1968 and 1975. However, in 1981 the Supreme Court said that a specific invention that was mostly a computer program was a process, not an algorithm, and thus patentable. After this, more software related patents flowed through the system and the Federal Circuit stated that a computer program must have a practical application, but this did little to limit patentability of most software. Though there has been no legislation on the legality of software patents, the failure of Congress to pass legislation has been interpreted as a sign that software patents are acceptable. [51] An on line petition has almost 12,000 signatures asking for a federal law forbidding software patents, but legislation has yet to materialize. [35] The only government response to public concern was a 1994 Arlington Public Hearings conducted by the PTO where various speakers from industry presented their views. No action was taken following this hearing. [38]

No major legislation specific to software patents has been pushed in the United States, but if passed, the Patent Reform Act of 2005 [H.R. 2795] will have some effects that may further disadvantage everyone involved in software patents. In order to be consistent with international

patent policy, this act switches the U.S. patent system from a “first to invent” to a “first to file” system. This means that regardless of any proof one has of invention prior to the issuance of a patent to someone else, the first one to acquire the patent has full rights to the invention. This disadvantages everyone as an individual could get a patent on something a company has been working on for years forcing the company to lose lots of money in research and licensing, and a company could get a patent on something an individual had spent their life working on. These problems apply to all kinds of patents but could be specifically damaging in the case of software.

Across the ocean, the European Union (EU) came up with a directive to allow for patenting of software. Originally the EU had the same vague policy on Software Patents that the U.S. does but their courts had interpreted this to mean that most software wasn't patentable. EU member states all had their own version of policy, many of which were against software patents. In March 2001, The UK Patent Office ended up with a strong statement against software patents:

> There is no sign, at least to date, of a want of innovation in computer-implemented business methods, and nor was there in the US before business methods became patentable in 1998. Intense innovation has characterised[sic] this field. The Government's conclusion is that those who favour[sic] some form of patentability for business methods have not provided the necessary evidence that it would be likely to increase innovation. Unless and until that evidence is available, ways of doing business should remain unpatentable. [42]

However, this proposed EU directive specifically allowed for software patents [33] and this created a global out roar: Huge petitions were created on the Internet including the Eurolinux petition with 434,960 signatures [36]; Linus Torvalds, Michael Widenius, and Rasmus Lerdorf, as mentioned above, appealed to the EU Council not to pass the directive. [2]; And various commissions and nations (Luxemberg for example [37]) performed independent studies on the subject, many of which found software patents to be a bad idea. As a result of this criticism, on

July 7, 2005 the EU voted to reject the directive and the Open Source world breathed a sigh of relief. [30] This does not mean that software will always be unpatentable in the EU, several software patents have already slipped through the examination process, but it does mean that the main venue for fighting software patents is back in the United States.

Conclusions

The concept of software patents has its proponents and opponents. Supporters feel like software patents work like any other kind of patent, furthering innovation while opponents feel that software patents work aggressively against innovation. The primary thing that both sides agree on is that some changes need to be made to most effectively work with software patents. In the Communications of the ACM article mentioned earlier, the author came to the following recommendations :

- Policy should be made on the assumption that innovation occurs in software as in other technologies until compelling evidence to the contrary is found.
- The PTO should be viewed as a source of innovation that competes for funding with other federally funded sources of innovation.
- The patent laws should be modified to make it possible for small entities to assert their patent rights more effectively.
- Further study of the role of patents and federal funding in software innovation is useful.[5]

The quandry is adequately summed up by Gregory Kirsch where he deduces that:

> [Assuming] that society wants to have wide access to innovative software, it is safe to say that society is willing to pay for this access, such as by allowing software developers to have proprietary rights in their creations. However, unlike the software developer, who wishes to have broad rights that allow him or her to reap a maximum return on investment, society likely is only willing to grant the software developer enough of an incentive so as to create a minimum threshold of innovation -- or usefulness -- in the developed software. In the end, society wishes to reward the software developer, but not to the extent that the developer would optimally desire.[10]

Many sources including a study by Carnegie Mellon University and California State University and a study by Research on Innovation and Boston University concluded that “Much remains to be understood about software patentability.” [49] and that changes to the existing system should not be made without more research and a good understanding of the results of any changes made. [46] Other conclusions include:

> “We don't know whether extending patent protection to computer programs and business methods implemented via computers will stimulate innovation in the software industry. [...] We should do more careful empirical research on the effects of increasing the availability of patents in high technology industries.” [55]

and:

> Overall, the patent system is working and should be improved rather than abandoned. There is no need to abolish patent protection for software-related inventions simply because some invalid patents may have been issued. Current and contemplated improvements to the patent system, as well as the checks and balances imposed by the federal courts, provide mechanisms to minimize the frequency of such occurrences and will remove many of these patents if they are granted in the future. [50]

Essentially the patent system is a working, developing, learning machine that has successfully encouraged innovation for years. Software is relatively new and the patent system is still figuring out how to deal with software related patents. It cannot be said that software patents are all good or all evil, only that some bad patents have made it though which have tarnished the public interpretation of the ability of the PTO to protect innovation. The logical next step for a policy change is to find ways to lower the barrier of entry for acquiring a patent, which may require legislative intervention, but case law over the next few years should take care of most of the problems. As companies fueled by Open Source Software, such as Red Hat, continue to grow , their coffers will slowly fill with the money required to go after companies with “bad” software patents and the system should eventually balance itself out. This will not be a quick path to a

perfect system, but there is no way to know the consequences of making drastic changes to the existing system and the outcome of hasty changes has the potential to do much more harm than good.

References

(All sources verified on December 4th, 2005)

[1] Adobe – Software Patent Policy
http://lpf.ai.mit.edu/Patents/testimony/statements/adobe.testimony.html

[2] Appeal to the EU Council, November 23, 2004
http://www.nosoftwarepatents.com/en/m/intro/app0411.html

[3] Confusion over software patents
http://www.mintz.com/images/dyn/publications/julian%20potter.pdf

[4] Corporate Statements on Software Patents
http://lpf.ai.mit.edu/Patents/testimony/statements/

[5] Debunking the Software Patent Myths, Communications of the ACM, June 1992
http://www.swiss.ai.mit.edu/6805/articles/int-prop/heckel-debunking.html

[6] GigaLaw.com: The Changing Roles of Patent and Copyright Protection for Software
http://gigalaw.com/articles/2000-all/kirsch-2000-04-all.html

[7] GigaLaw.com: How the Patent Office Has Responded to E-Commerce Patents
http://gigalaw.com/articles/2000-all/kirsch-2000-08-all.html

[8] GigaLaw.com: Lessons from the United States and Europe on Computer-Related Patents
http://gigalaw.com/articles/2001-all/mccoy-2001-08-all.html

[9] GigaLaw.com: Patent Lessons Provided by PayPal and Palm
http://gigalaw.com/articles/2002-all/isenberg-2002-02-all.html

[10] GigaLaw.com: Software Protection: Patents versus Copyrights
http://gigalaw.com/articles/2000/kirsch-2000-03.html

[11] GigaLaw.com: The Software and E-Commerce Patent Revolution
http://gigalaw.com/articles/2000-all/kirsch-2000-01-all.html

[12] The Grokline Patent Project
http://www.grokline.net/pblist.php

[13] IBM Pledges 500 U.S. Patents To Open Source In Support Of Innovation And Open Standards
http://www-

1.ibm.com/press/PressServletForm.wss?TemplateName=ShowPressReleaseTemplate&SelectString=t1.docunid=7473

[14] IBM – Software Patent Policy
http://lpf.ai.mit.edu/Patents/testimony/statements/ibm.testimony.html

[15] IBM, Sony, Philips form Linux alliance
http://money.cnn.com/2005/11/10/technology/linux.reut/index.htm

[16] Intel – Software Patent Policy
http://lpf.ai.mit.edu/Patents/testimony/statements/intel.testimony.html

[17] ITT: May 2001 – IPR for software: Patent Pending?
http://www.cordis.lu/itt/itt-en/01-3/policy05.htm

[18] Julie E. Cohen & Mark A. Lemley, Patent Scope and Innovation in the Software Industry
http://www.law.berkeley.edu/journals/clr/library/cohen-lemley01.html

[19] The Letters of Thomas Jefferson: To Issac McPherson Monticello, August 13, 1813
http://odur.let.rug.nl/%7Eusa/P/tj3/writings/brf/jefl220.htm

[20] Microsoft – Software Patent Testimony
http://lpf.ai.mit.edu/Patents/testimony/statements/microsoft.testimony.html

[21] Moore's Law
http://en.wikipedia.org/wiki/Moores_law

[22] Microsoft Bows to Eolas, Revamps IE's Multimedia Handling
http://www.eweek.com/article2/0,1895,1895907,00.asp

[23] Microsoft's Public Policy Goals: Help Foster Growth and Innovation
http://www.microsoft.com/presspass/features/2005/aug05/08-30PublicPolicy.mspx

[24] Negative Correlation of Innovation and Software Patents
http://www.base.com/software-patents/scoreboard.html

[25] Novell: Patent Policy
http://www.novell.com/company/policies/patent/

[26] Open invention network formed to promote Linux and spur innovation globally though access to key patents
http://www.openinventionnetwork.com/press.html

[27] Open-source honchos trash software patents

http://news.com.com/Open-source+honchos+trash+software+patents/2100-7344_3-5559647.html

[28] Open Source Software
http://en.wikipedia.org/wiki/Open-source_software

[29] Oracle Corporation – Patent Policy
http://lpf.ai.mit.edu/Patents/testimony/statements/oracle.statement.html

[30] Patentability of computer-implemented inventions
http://europa.eu.int/comm/internal_market/en/indprop/comp/index.htm

[31] Patents are killing software innovation
http://news.zdnet.com/2100-3513_22-5342291.html

[32] The Patent Busting Project, Electronic Frontier Foundation
http://www.eff.org/patent/

[33] Patents: Commission proposes rules for inventions using software
http://europa.eu.int/comm/internal_market/en/indprop/comp/02-277.htm

[34] Path to software innovation leads away from patents
http://searchopensource.techtarget.com/originalContent/0,289142,sid39_gci1115112,00.html

[35] Petition Against Software Patents
http://www.petitiononline.com/mod_perl/signed.cgi?pasp01

[36] Petition for a Software Patent Free Europe
http://petition.eurolinux.org/signatures.html

[37] Position of the innovation platform regarding EU software patent directive
http://libre.tudor.lu/SwPat

[38] Public Hearing on Use of the Patent System to Protect Software-Related Inventions
http://www.uspto.gov/web/offices/com/hearings/software/arlington/vahrng.pdf

[39] Red Hat exec criticizes software patents, Microsoft
http://www.infoworld.com/article/05/04/19/HNrhmysql_1.html

[40] Red Hat, Inc. Statement of Position and Our Promise on Software Patents
http://www.redhat.com/legal/patent_policy.html

[41] Richard Stallman: Software Patents, Cambridge, UK, 2002-03-25
http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html

[42] Should Patents be Granted for Computer Software or Ways of Doing Business?
http://www.patent.gov.uk/about/consultations/conclusions.htm

[43] Silicon Graphics – Software Patent Testimony
http://lpf.ai.mit.edu/Patents/testimony/statements/sgi.testimony.html

[44] Recent Developments In Software Patents
http://www.cla.org/soft%20patent.pdf

[45] Software Development Industry Study, January 2001
http://www.sbtdc.org/pdf/software.pdf

[46] The Software Patent Experiment, March 16, 2004
http://www.researchoninnovation.org/softpat.pdf

[47] Software Patents from the Inside
http://www.tbray.org/ongoing/When/200x/2003/09/15/SWPatents

[48] Software Patents in Europe, Free Software Foundation Europe
http://www.fsfeurope.org/projects/swpat/swpat.en.html

[49] Software Patents: Innovation or Litigation?
http://www.sei.cmu.edu/programs/acquisition-support/publications/sw-patents.pdf

[50] Software Patents: Just Make A Good Thing Better
http://www.mttlr.org/voltwo/syrowik.pdf

[51] Software patents under United States patent law
http://en.wikipedia.org/wiki/Software_patents_under_United_States_patent_law

[52] Software Patents: What does “Means” Mean?
http://library.findlaw.com/2004/May/11/133414.html

[53] TechWeb: News: Torvalds, Open-Source Pioneers Take On EU Patent Issue
http://www.techweb.com/wire/software/54200078

[54] Vendors Pledge to Share Linux Patents
http://news.yahoo.com/s/pcworld/20051110/tc_pcworld/123486

[55] You Can Patent That?
http://www.phil.frb.org/files/br/brq101bh.pdf

Appendix I: Mentioned Individuals

Mitch Kapor – Founder of Lotus Development Corporation and the designer of Lotus 1-2-3 (The "killer app" in the business world in the 1980s and early 1990s), Mitch led Lotus to a revenue of $156 million in 1984. He co-founded the Electronic Frontier Foundation, founded the Open Source Applications Foundation, and has been the chair of the Mozilla Foundation since its inception in 2003. (http://en.wikipedia.org/wiki/Mitch_Kapor)

Rasmus Lerdorf – A native of Greenland, Rasmus is the author of the first version of the PHP language which is now the most popular scripting language on the Internet and used to power sites such as Yahoo! Inc. where Rasmus is now employed as an Infrastructure Architecture Engineer. According to Netcraft, an Internet statistics gathering company, as of October 2005 there are almost 24 million web sites running PHP.
(http://en.wikipedia.org/wiki/Rasmus_Lerdorf, http://www.php.net/usage.php)

Linux Torvalds – Most well known for starting the development of Linux, this Finnish software engineer is the "Benevolent Dictator for Life" of Linux which is now seen as a threat by companies as large as Microsoft. He is now employed by the Open Source Development Labs in Oregon, and continues to focus on the Kernel (the "Core") of Linux. Linux runs everything from toasters to web servers and runs much of the technologies that power the Internet. The Linux Counter estimates that there are 29 million Linux users.
(http://en.wikipedia.org/wiki/Linus_Torvalds, http://counter.li.org/)

Michael Widenius – Michael is the main author of the first version of MySQL, an open source database application with an estimated six million installations. He is the chief technical officer of MySQL AB (the company that sells support and binaries of MySQL) and is still an important force in the development of MySQL (http://en.wikipedia.org/wiki/Michael_%28Monty%29_Widenius)

Appendix II: Mentioned Companies

Adobe - “Adobe today is one of the world's largest software companies, generating annual revenues exceeding U.S. $1.6 billion. Approximately 4,000 employees around the world share Adobe's commitment to helping people and organizations communicate better. Headquartered in San Jose, California, Adobe is traded on the Nasdaq National Market under the symbol ADBE.” (http://www.adobe.com/aboutadobe/main.html)

Autodesk - “Today, Autodesk is a fully diversified software company that provides targeted solutions for creating, managing, and sharing digital assets. Our community numbers more than six million users and includes four global strategic partners—Microsoft, Intel, Hewlett-Packard, and IBM—as well as 2,500 third-party developers. What unites us all is the same spirit our company was founded on: innovation that drives real-world success. You can see that success in the accomplishments of our loyal customers, which include 100 percent of Fortune 100 companies and 98 percent of Fortune 500 companies.” (http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=5359166)

Borland - “With more than 1,300 employees worldwide and operations in more than 20 countries, Borland has served over 80 percent of global 2000 companies with best-in-class solutions for more than 22 years. Borland is committed to the ongoing delivery of software, services, and partnerships to help our customers cost-effectively transform their software delivery processes, with minimum risk and maximum ROI.” (http://www.borland.com/us/company/index.html)

IBM - “At IBM, we strive to lead in the invention, development and manufacture of the industry's most advanced information technologies, including computer systems, software, storage systems and microelectronics.” IBM has over 329,000 employees with assets totaling over $109 billion and annual revenue exceeding $96 billion. (http://www.ibm.com/ibm/us/)

Intel - “At Intel, we believe in innovation. We're driven by it. We live by it. And it's this principle that led us to create the world's first microprocessor back in 1971.” Founded in 1968, Intel currently has 91,000 employees, has 294 facilities across the globe and it's annual revenue exceeds $34.2 billion. (http://intel.com/intel/index.htm?iid=HPAGE+low_about_aboutintel&)

Microsoft – Founded in 1975, Microsoft has 63,564 employees in almost 100 different countries and a annual revenue of almost $40 billion. (http://www.microsoft.com/presspass/inside_ms.mspx) The majority of desktop computers run Microsoft's Windows operating system and Microsoft's Office is the de facto standard for office productivity tools.

Novell - Founded in 1983, “Novell serves customers in varied market segments, who highly value interactive business solutions in the data center, security and identity, resource management, workgroup, and desktop spaces.” Their annual revenue is over $1.1 billion and they employ more than 5,000 people. (http://www.novell.com/company/fastfacts.html?sourceidint=t2_about_novell:%20Fast%20facts)

Oracle - “Today Oracle (Nasdaq: ORCL) is still at the head of the pack. Oracle technology can be found in nearly every industry around the world and in the offices of 98 of the Fortune 100 companies. Oracle is the first software company to develop and deploy 100 percent internet-enabled enterprise software across its entire product line: database, business applications, and application development and decision support tools. Oracle is the world's leading supplier of software for information management, and the world's second largest independent software company.” (http://www.oracle.com/corporate/story.html)

Phillips Electronics - “Royal Philips Electronics is one of the world's biggest electronics companies and Europe's largest with sales in 2004 of Eur 30.3 billion. Active in over 60 businesses, and with more than 115,000 registered patents, Philips is currently number 1 in the global markets for lighting, electric shavers and DVD recorders. And we're number 2 in medical diagnostic imaging worldwide. Within the cyclical goods market, Dow Jones recently ranked us the global leader in sustainability.” (http://www.philips.com/about/company/article-14054.html)

Red Hat - “Founded in 1993, Red Hat is the premier Linux and open source provider. The most recognized Linux brand in the world. We serve global enterprises through technology and services made possible by the open source model. Solutions include Red Hat Enterprise Linux operating platforms, sold through a subscription model, and a broad range of services: consulting, 24x7 support, Red Hat Network. Red Hat's global training program operates in more than 60 locations [with almost 1000 employees] worldwide and features RHCE, the global standard Linux certification.”

(http://www.redhat.com/en_us/USA/home/company/companyprofile/facts/)

SGI – SGI provides high performance computing and graphics solutions for Government and Defense, Science, Manufacturing, Energy, and Media markets. With 2400 employees they lead industry in R&D investment with 14% of total revenue from the past 20 years being invested in research and have an annual revenue of $730 million.
(http://www.sgi.com/company_info/newsroom/factsheet.html)

Sony - “Sony is a leading manufacturer of audio, video, communications, and information technology products for the consumer and professional markets. Its music, motion picture, television, computer entertainment, and online businesses make Sony one of the most comprehensive entertainment companies in the world. Sony's principal U.S. businesses include Sony Electronics Inc., Sony Pictures Entertainment, Sony Computer Entertainment America Inc., and a 50% interest in SONY BMG Music Entertainment, one of the largest recorded music companies in the world. Sony recorded consolidated annual sales of approximately $67 billion for the fiscal year ended March 31, 2005, and it employs 151,400 people worldwide. Sony's consolidated sales in the U.S. for the fiscal year ended March 31, 2005 were $18.4 billion.”
(http://sony.com/SCA/corporate.shtml)

Sun - “Sun's products include computer servers and workstations based on its own SPARC and AMD's Opteron processors, the Solaris and Linux operating systems, the NFS network file system, and the Java platform.” (http://en.wikipedia.org/wiki/Sun_Microsystems) In the first

quarter of fiscal 2006, Sun had a net revenue of almost $3 billion. (http://biz.yahoo.com/e/051104/sunw10-q.html)

Synopsis – "Synopsys, Inc. (Nasdaq:SNPS) is a world leader in semiconductor design software, developing software and offering professional services that companies use to design systems-on-chips (SoCs) and electronic systems. The company sells its products to semiconductor, computer, communications, consumer electronics, aerospace and other companies that develop electronic products." Founded in 1986, Synopsis has over 4,000 employees and an annual revenue of over $1 billion. (http://www.synopsys.com/corporate/co_profile.html)

Taligent - "Taligent was the name of an object-oriented operating system and the company dedicated to producing it. Today, both are gone." "Taligent [...] licensed various technologies to Sun which today are part of Java, as well as to Oracle and Netscape. After two years as a wholly-owned subsidiary, Taligent was dissolved in January 1998 and the engineering teams became IBM employees." (http://en.wikipedia.org/wiki/Taligent)

Time Warner - "Time Warner Inc. is a leading global media and entertainment company with businesses in filmed entertainment, interactive services, television networks, cable systems and publishing. Whether measured by quality, popularity or financial results, our divisions are at the top of their categories. America Online, Time Inc., Time Warner Cable, Home Box Office, New Line Cinema, Turner Broadcasting System and Warner Bros. Entertainment maintain unrivaled reputations for creativity and excellence as they keep people informed, entertained and

connected." "We are innovators in technology, products and services. On the forefront of technology, Time Warner has pioneered industry-shifting products such as the DVD and digital cable. And Time Warner's businesses continue to develop new and uniquely effective ways for our partners and advertisers to reach the markets, audiences and customers they seek." (http://www.timewarner.com/corp/aboutus/our_company.html)

Wind River Systems - "Wind River is the global leader in device software optimization (DSO). We enable companies to develop and run device software better, faster, at lower cost, and more reliably. Wind River technology is currently deployed in more than 300 million devices worldwide by industry leaders like Apple, Hewlett Packard, Boeing, Motorola, NASA, and Mitsubishi. Wind River Professional Services enable leading electronics vendors like Philips, Siemens, Nortel, and Samsung to design, develop, and deploy innovative products on or ahead of schedule, at or below budget."