

Why Software Developers Should Support a New, Limited Patent

Lee A. Hollaar

Professor, School of Computing
University of Utah

May 3, 2007 version

Copyright © 2007

This paper is licensed under the
Creative Commons Attribution-Share Alike 2.5 License

(This paper was first presented at EUPACO-2,
The European Patent Conference, Brussels, 15-16 May 2007.
The latest version of this paper, as well as the paper
on which it is based, can be found at
<http://digital-law-online.info/papers/lah/mini-patent.htm>
Comments or suggestions should be sent to hollaar@cs.utah.edu)

Introduction

Suggesting patent protection to free or open-source software developers is like waving a red flag in front of a bull.¹ The reason for this reaction on the part of software developers is likely a combination of not fully understanding patents, problems caused by the limited examination that is currently given a patent application in the United States (examiners spend about twenty hours or less on the average from the start of the examination to a final disposition), and a religious fervor against patents on the part of some, in part because patents can stop the “freeing” of a proprietary program like Unix.²

The way patents are currently seen has obscured one of their primary benefits – the disclosure of technical knowledge that might otherwise remain as a trade secret,³ “in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same.”⁴

¹ When somebody posted an item on Slashdot about my article in the February 2006 *IEEE Spectrum*, “Patents 2.0: A new type of patent is needed” (<http://www.spectrum.ieee.org/feb06/2785>), there were more than a hundred-fifty comments, most from people viscerally reacting to “patents” without having even read the article or the longer paper it referenced.

² See, for example, United States Patent 4,135,240, “Protection of data file contents,” granted on January 16, 1979, to Dennis Richie and assigned to Bell Labs. This patent covers the operation of the SUID bit in Unix, and any reimplementations of Unix would likely infringe it. After the patent was granted, Bell Labs dedicated the invention to the public.

³ The very word “patent” comes from “open.” The full name of a patent is “Letters Patent” or “open letters” and refers to grants from a sovereign that were open for public inspection

⁴ 35 U.S.C. § 112, first paragraph.

I have proposed a new, limited patent. It would not replace current utility patents in the United States and other countries, but may be a better alternative than the current utility model protection in some countries. And by providing an alternative option with easier, faster, but limited protection, it will help to eliminate the backlog of about a million pending patent applications in the United States, many for computer or communications techniques, and allow for a better examination.

Details of my proposal can be found in my evolving paper “A New Look at Patent Reform.”⁵ In this paper, I discuss why what I propose addresses the concerns of software developers regarding current patents and provides substantial benefits, both to free and open source developers and to proprietary software companies.

Patent protection: too much, too long, too late

In my paper, I discussed the problems with patent protection for software and other fast-moving technologies:

- Patent protection often goes beyond what is needed to prevent competitors from usurping new techniques, with protection lasting about two decades and blocking those who independently created the technology.
- It takes too long to get patent protection, particularly for fast-moving technologies that can be readily copied once they are being used.
- Because of the requirement for nonobviousness, it should be difficult to get a patent, but the limited examination dictated by current application fees often doesn't give the examiner time to find and consider important prior art.

There are efforts going on to try to reduce the delay between the filing of a patent application and receiving the first office action on the merits of that application, which is now reaching three to four years in some software-related areas.⁶ But as seen with the advent of “patent trolls,” the broad scope of protection of a patent is often not commensurate with the depth of the disclosure.

And while better examination is something that everybody says is needed, there is little consideration that paying for a substantially better examination will require a corresponding substantial increase in application fees, resulting in fewer applications and therefore fewer organized descriptions of the prior art going into the patent office's database. And that database not only helps examiners easily locate prior art for pending applications, but provides the public an organized and evolving source of information about technology.

Isn't copyright sufficient?

Most software developers seem to think that copyright protection is sufficient, perhaps augmented by licenses that require specific behavior in trade

⁵ <http://digital-law-online.info/papers/lah/mini-patent.htm>.

⁶ The United States Patent and Trademark Office is trying to hire 1,200 patent examiners this year, and every year in the foreseeable future, both to try to reduce the pendency and to replace examiners lost to law firms. But it is unlikely that the USPTO can hire its way out of its problems.

for being able to use, modify, or redistribute the software.⁷ But copyright only protects the expression of a technique, and not the technique itself. If a competitor can determine the method of a computer-based invention and implements it without reproducing its copyrighted expression (such as producing a “clean-room” implementation based on a functional description), there is no copyright infringement. Often, a technique is self-revealing, so that once competitors are aware of it, it is not difficult for them to incorporate it into their products or services.

This is a special problem when the source code is publicly available, since it would be easy for a company that didn’t want to comply with the license that accompanies the source code to study it to learn how it does things and then describe that to a programmer who has not seen the source code to include it as part of its proprietary program. For example, if a new technique in an open source program substantially improved the performance of a relational database system, there would be little to keep companies that develop proprietary relational database systems from using it, even if they did not make their own source code available.

This one-way transfer of new techniques from open-source developers to proprietary software companies will only grow more acute as open-source programmers go from reimplementers trying to “free” proprietary programs to innovators creating new and unobvious (and therefore patentable) technology, but decline to file for patent protection.

A lesson from the past

It might seem appealing to extend copyright to protect in such instances. But experience shows that that will cause more problems than it solves. Before it became clear through a series of court decisions that software-based inventions were patentable, we had gone well beyond literal copying as infringement to protecting the “non-literal” aspects of the program.

The high-water mark in non-literal copyright protection for computer software came in *Whelan v. Jaslow*,⁸ which held that the “structure, sequence, and organization” of a computer program was protected by its copyright. In essence, the court addressed the boundary of what was copyrightable expression and what was an uncopyrightable idea by finding that the overall purpose of the program (in this case, running a dental lab) was the idea and anything used to implement that idea was protected expression. That included file structures, screen displays, and the functionality of similar subroutines.

We don’t know how far courts would have continued to stretch copyright beyond literal infringement because about the time *Whelan* was decided, the Supreme Court had found an algorithm-based invention that it felt was patentable⁹ and the Federal Circuit had completed its embrace of software

⁷ But if a license goes too far beyond statutory copyright, it may be viewed as misuse and the copyright becomes unenforceable in court. The leading case in this regard is *Lasercomb America v. Reynolds*, 911 F.2d 970, 15 USPQ2d 1846 (4th Cir. 1990), which I discuss at <http://digital-law-online.info/lpdi1.0/treatise15.html#secII.K>.

⁸ 797 F.2d 1222, 230 USPQ 481 (3d Cir. 1986).

⁹ *Diamond v. Diehr*, 450 U.S. 175 (1981). But software patents had issued well before then. For example, see U.S. Patent 3,568,156, “Text Matching Algorithm,” granted in 1971. (The inventor, Kenneth Thompson, is also one of the creators of the Unix operating system.)

patents with *In re Alappat*.¹⁰ As software patents became the preferred means for protecting a new technique, copyright reverted to protecting against the literal copying of a computer program.

But even if copyright could be used to protect software techniques, there will be two problems. The first is the term of protection. Many people think the twenty-years-from-filing term for patents is far too long for computer software. Patents for Microsoft's Windows 95 are now just expiring. But copyright lasts seventy years beyond the death of the last author or, in the case of a published work made for hire, 95 years. The copyright on Windows 95 will not expire until the end of 2090!

Disclosure and claiming is important

The second problem is the lack of a disclosure requirement in current copyright law. As mentioned above, the disclosure requirement forms an important part of the patent system, although it is used by far too few software developers.¹¹ Even with "open source" software, it is difficult to find how a particular function is performed unless that function is an obvious part of a known program.

In fact, since adoption of the Copyright Act of 1976, there is no longer a requirement that the protected work even be published. A trade secret, written down or otherwise fixed in a tangible medium of expression, is protected to the same extent as a book on sale,¹² even though its protected expression is unavailable except through a trade secret agreement. This is the case for most proprietary computer software.

In contrast, a patent concentrates on one particular technique, and that technique must be described fully in the published patent, so that a skilled person can implement and use the technique without undue experimentation. The disclosure is also manually placed within a classification system so that it can be readily located.¹³

A copyright comes into being at the time of fixation of a work, and a simple registration form must be filed before an infringement suit can be brought. But such simplicity comes at a price – as cases like *Whelan* show, it is hard to determine just what is protected by a copyright, making it difficult for a person wanting to produce a new implementation of a computer program. Because of the claiming requirement for patents, it is far easier to know in

¹⁰ 33 F.3d 1526, 31 USPQ2d 1545 (Fed. Cir. 1994).

¹¹ There is a myth that if one is aware of a patent and infringes it, the damages will be tripled. 35 U.S.C. § 284 really provides for increasing the damages by "up to three times" (emphasis added) in any case, with no special provision for willful infringement as in United States copyright law, although it is more likely that the judge will increase the damages when the infringement is willful.

¹² Perhaps even more, since the term of a work made for hire is 95 years from its first publication, or 120 years from the date of its creation, whichever ever comes first. 17 U.S.C. § 302(c).

¹³ When there are too many patents within a particular class and subclass, the patent office breaks the subclass (and related subclasses) into more specific subclasses or sometimes a new classification. For example, software-based inventions were initially a subclass within the class for computers. They later became their own class. Now, they span a number of classes, with an entire class for database techniques and another for artificial intelligence.

advance what a patent covers than what a copyright covers, especially if copyrights were to expand again to cover more and more non-literal aspects of a computer program because patent protection is not longer available.

Intermediate forms of protection

Eliminating software patents and going to copyright as the only protection is likely to cause new distortions in copyright. Similarly, the solution to the problems with patents will not be found by fine-tuning the current patent statutes and rules. It is better to look at those aspects of copyright protection, such as the defense of independent creation, and combine them with the best parts of patents. Such protection could be used in lieu of a patent, or until a patent is granted.

The United States took a small step in that direction with the passage of legislation protecting “useful articles,” although the only article that Congress seems to feel is useful is a “vessel hull.”¹⁴ Others have proposed special protections for “useful articles,”¹⁵ but those laws or proposals are generally limited to protecting mechanical devices and other manufactured items. They do little or nothing to protect software-based inventions, methods of doing business, or other processes, which as discussed above are areas where an intermediate form of protection may be the most useful.

A number of countries have a “petty patent” or “utility model” of intermediate protection.¹⁶ Unlike a regular patent, the intermediate forms of protection generally have no substantive examination before issue, a shorter term of protection, and a lower threshold of “inventiveness.” While Germany does not allow the protection of processes and methods under its utility model, it may be possible to protect computer programs and even business methods with clever claiming, such as claiming apparatus, articles of manufacture, or signals as is now common in the United States for software-based inventions.

The United States is starting to discuss similar ideas. The *United States Patent and Trademark Office 2007–2012 Strategic Plan* states:

A longer-term endeavor, critical to addressing quality and timeliness, is working with our stakeholders, the Administration, Congress, and our international partners to determine if there is some combination of examination alternatives that will better meet applicants’ needs while providing a more efficient use of USPTO examination resources.

But there is a problem with the utility model. Even though there has not been a substantive examination of the application, others may not make or sell the protected item. Even if prior art is known that would invalidate the filed claims when examination is requested, that may only result in narrowed claims that avoid the prior art but still cover what one is doing. Until examination has been completed, utility model protection can cause substantial uncertainty for somebody developing a similar product.

¹⁴ See 17 U.S.C. § 1301(b)(2).

¹⁵ One group that has been advocating article protection for decades is IEEE-USA. <http://www.ieeeusa.org/forum/POSITIONS/newip.html>.

¹⁶ These include Japan, China, Taiwan, and most European countries with the exception of the United Kingdom, Sweden, and Luxembourg. In May 2001, Australia introduced a new alternative patent, which it calls an “innovation patent.”

With the Supreme Court's unanimous decision in *KSR v. Teleflex*,¹⁷ making it easier to show that a claim is obvious in light of the prior art, it will become more difficult to get a patent – in some cases, because the invention doesn't warrant it, but in other cases because the examiner has rejected the claims in a hindsight reconstruction – or take more time to get one. This will make an alternative form of protection more attractive in the United States.

What to protect?

So, the question is, what shape should such a new limited patent take?

While more details of my proposed limited patent protection can be found at <http://digital-law-online.info/papers/lah/mini-patent.htm>, here is a brief overview of what I am proposing:

- **How creative?** Novelty may be the most appropriate standard for this intermediate protection. This would prevent people from receiving protection for things that are already available to the public while avoiding the problems associated with trying to prove or disprove that something is “obvious” in light of the prior art. But there should also be a limited form of nonobviousness required: that if a process or method is prior art, simply implementing that process or method on a computer or storing it on some medium does not result in a novel invention unless there is something nonobvious in the way the computer is being used or the information is being stored. No patentable weight should be given to the use of a computer or storage on a commonly-available medium should also be considered.
- **Requirements for protection.** Since we are trying to provide more timely protection than patents, it is unreasonable to delay the protection during an examination period. But simply providing protection based on the marking of an item or a simple registration does not give sufficient notice to the public of the aspects of the item that are protected.
- **When protection comes into being.** Protection could come about by first filing of an application as the protected item becomes close to public availability, along with a nominal filing fee, around \$500. The protection would come into being when the technology is first used in commerce within the United States, marked with its application registration number. Since the primary purpose of this limited patent is to provide immediate protection for technology that could be copied by a competitor before patent protection could be secured, there is no need to provide the protection for speculative inventions or those not available to the competitors.
- **Disclosure requirement.** One of the important aspects of patents is their disclosure of the invention, not only to the public but for the prior art collection used to examine later patent applications. It is important that this intermediate protection be given only after its technology is adequately described. The limited patent proposed in this paper would continue this disclosure requirement and the classification of techniques by the patent office, and would enhance it by encouraging more filings because of the lower fee and simplified registration procedure.

¹⁷ <http://www.supremecourtus.gov/opinions/06pdf/04-1350.pdf>.

- **Claiming.** A claiming requirement should be a part of any intermediate protection. But speculative claims should not be allowed. To be valid, a claim must cover at least one of the registrant's products. There is no need for a "doctrine of equivalents" for claim elements, since technology unforeseeable at the time of the application will not be in a registrant's product and therefore would not be protectable. This will make it easier to determine the effective scope of a claim.
- **No initial examination.** No examination beyond a check to see that the requirements for registration were met would be performed. In particular, there would be no search to determine if the protected technology was, in fact, novel. Because there would be no administrative determination of novelty, there would be no presumption of validity for the limited patent.
- **Priority.** Probably one of the most contentious issues in United States patent reform is whether a patent should be granted to the first person to invent or the first inventor to file. For this intermediate form of protection, it may make little difference. With current utility patents, the patent owner has the right to collect royalties from, and even stop the practice of the invention by, a person who independently developed the claimed invention. However, because substantial independent creation would be a defense to infringement, there should be no concern about that. While it would not be a problem even if both independent developers received protection, if it were desirable to have only a single patentee, it could be given either to the first one to file for registration or, as a way to promote prompt commercialization of the protected technology, the first to market in the United States.
- **Ownership.** While in the United States a patent is nominally owned by the inventors, most employees and many contractors have signed agreements requiring the assignment of the application and any resulting patents to the company. Rather than add to the application expenses by requiring such formalities, it would be better to follow the "made for hire" rules as in copyright.¹⁸ This would be limited to things within the scope of the employment, so that inventions in other areas would be owned by their creators if they can put them in commerce.
- **Length of protection.** Four years seems like a reasonable term. It would provide a head start against competitors, which could be maintained by developing new features for the technology during the four-year term of protection, each of which could qualify for its own limited patent with a new four-year term. Of course, like any patent protection, when the patent expires anybody can market the patented technology, but not improvements protected by subsequent patents until those patents expire. This encourages the development of improvements to patented technologies. Having such a relatively short term would also reduce the impact of the protection on the aftermarket parts sector as well as fast-moving technologies.
- **What acts violate the protection?** Making or selling, but only against those who were aware of the technology and its protected status, not independent creators. Indirect liability, primarily inducement to violate

¹⁸ See 17 U.S.C. §§ 201(b) and 101.

the protection, would address those who sponsor the production of things that violate the protection. While it is common in United States patents to use claims in a variety of classes so that people selling the software will be direct infringers, it would be far better to make such people infringers under the statute, much like other indirect infringers are,¹⁹ so that *Beauregard*-type claims²⁰ would not be necessary and the problems they can cause would be avoided.

- **Defenses.** Showing prior development and use would be an absolute defense, and also invalidate the limited patent, since novelty would be a requirement for the protection. Substantial completion would also be a defense, but it would be a personal defense, not usable by another party. The burden of proof would be on the alleged infringer to show substantial completion by development logs or other documentation.
- **Later examination.** Examination could be requested by any party by the payment of the examination fee, along with the submission of any relevant prior art known to the party. Claims could not be broadened during this examination, so the initial claims would provide a high-water mark for the eventual protection. After examination, the limited patent would be presumed valid with respect to the prior art considered. This would be an inexpensive way for a potential defendant to invalidate the limited patent if it were not novel.
- **Prelitigation requirements.** Because there would be no examination required before protection begins, examination should occur as an initial step in any litigation. This would reduce the cost of litigation when a violation is alleged and provide for a faster resolution of the novelty issue. At the time an infringement suit is filed, the proceeding would be stayed pending a novelty examination by the patent office. This would be less time consuming than a regular patent examination, because nonobviousness would not be considered and the alleged infringer would be aware of the examination (because of the suit) and be able to provide prior art for the examiner to consider. This will shift the determination of novelty from a judge or jury, inexperienced in the particular technology, to the patent office and its technically-trained examiners.

¹⁹ 35 U.S.C. § 271 addresses a variety of indirect infringements: inducement in subsection (b), contributory infringement in subsection (c), supplying components for foreign manufacture in subsection (f), and manufacturing in a foreign country using a patented process in subsection (g).

²⁰ Named after the inventor whose application IBM used to get the courts and the USPTO to accept claims to a “computer program product” comprising a medium storing a computer program that implements a patentable method. By using such claims, those creating or distributing the media for a computer program become direct infringers, avoiding the possibility that they might escape through one of exceptions for contributory infringement or inducement. But as software distribution goes from floppy disks and CDs to the Internet, such claims become ineffective. To counter that, some have proposed claiming signals that carry patented software, but that could make an ISP an infringer when it “makes” the signal as it routes the data.

Addressing software patent concerns

In my proposal, the purpose of this new protection is to protect against somebody “knocking off” a product in commerce. Because of that, it is possible to provide a strong defense for independent creation as well as partially address “patent trolls” by requiring that there be a product actually being delivered in commerce for there to be protection.

There is a reason why software developers are less concerned about copyrights than patents, including their longer term. To infringe a copyright, you have to have based your work on the copyrighted work. No matter how similar your work is to another, if you can show that you independently created your work, you are not an infringer.

There is no such safe harbor for a patent infringer. If what you are doing meets all the elements of any claim of a patent, you are an infringer. It makes no difference whether you have ever seen the patented thing or are aware of the patent. As some recent high-profile cases have shown, a software developer can plow millions into development of a new system, but can be stopped by the owner of a patent that is not even producing a product or licensing the technology to a manufacturer.

This scares most software developers, especially when the quality of some patents is considered. But protection for independent creators, even when they have not fully completed their own work, should address that concern.

Because of the requirement that notice of the protection be included with the product, it would be simple for a software developer to know all the limited patents protecting a software product. An examination of the claims would give the metes-and-bounds of the protection, so that infringement could be avoided.

But more importantly, because infringement requires derivation from the protected article in commerce, if one is not trying to clone or develop a program similar to an existing one, there is no need to worry about infringement.

Better examination for regular patents

One problem not discussed by the advocates of better examination for regular patents is the effect on fees. A better examination will necessitate an increase in fees, perhaps a substantial one, to pay for the increased time spent by the examiner reviewing prior art and addressing the arguments of the applicant, as well as a “second pair of eyes” review as is now the case for business method patents,²¹ if that idea were extended to other arts. But even though patent application fees are a small part of the cost of filing for a patent,²² any substantial increase will likely lead to a reduction in patent filings, especially by cash-strapped small companies and inventors.

We saw the effect of discouraging the filing of applications when the USPTO’s policy was not to grant patents on software-based inventions, or at least make it difficult for applicants to get such a patent. Software developers didn’t file applications on their advances because they didn’t believe that patent

²¹ See <http://www.uspto.gov/web/offices/com/sol/actionplan.html>.

²² The current application fee is \$770, and is half that for “small entities.” (That obviously doesn’t pay for a lot of examiner time.) If the patent is allowed, there is a \$1330 issue fee, also discounted by fifty percent for small entities. In contrast, patent attorneys may charge \$5000 or (often) more to prepare a patent application.

protection was available, resulting in a gap in the USPTO's prior art collection corresponding to the formative years of software systems²³ and a stretching of copyright protection by the courts to fill the gap left by not having patent protection available.²⁴ We are still paying for that gap in the prior art collection in terms of patents being issued on old techniques, and we cannot afford to have that happen again.

By providing an intermediate form of patent protection, it would be possible to provide the examination a regular patent deserves without leaving things unprotected or reducing disclosures of the prior art. Instead of a single examiner spending about twenty hours on an application, an increase in application fees by a factor of twenty²⁵ could pay for a team of examiners, led by a senior examiner, spending on the order of 400 hours searching prior art and assessing the obviousness of an invention.

Such a fee increase could also reduce the workload on the patent office since many inventors might opt for the lower-cost, and immediate, protection of the limited patent and not go for a full patent.

But of more importance, because there is an alternative form of protection available, the requirement for granting a full-fledged utility patent, with its long term and winner-take-all approach, can go from "A person shall be entitled to a patent *unless*"²⁶ to "A person shall be entitled to a patent *if*."

Conclusion

While a number of reports have made suggestions for improving the United States patent system, improving the quality of examination may have unexpected consequences. The increased examination fees may discourage the filing of patent applications, thereby hurting the prior art collection needed to properly examine applications. Heightened scrutiny for nonobviousness will likely increase pendency, particularly for inventions where patents are the only available form of protection and so the applicant must continue prosecuting an application until a patent is granted.

For fast-moving technologies, current patent protection is too much, too long, and too late. The creation of a limited patent could provide the necessary protection while allowing substantial improvements to the quality of the examination of regular patents.

And it would provide software developers, especially those creating open-source software, with a way of protecting their new technology from commercial

²³ A personal example: In 1969, working for the Chicago software company Datalogics, I developed a new way for composing complex, multicolumn page (such as the yellow pages) and producing an output for a phototypesetter that only required forward motion of the film. The technique was at least ten times faster than other systems, and allowed Datalogics to become a leader in computer typesetting systems. (At one time, about two thirds of law reviews, for example, were composed using Datalogics software.)

The technique remained a trade secret of Datalogics, since copyright would not protect the technique itself and patents seemed unavailable. As far as I know, a description of the technique has never been available to the public and so the technique has been essentially lost.

²⁴ See, for example, *Whelan v. Jaslow*, 797 F.2d 1222, 230 USPQ 481 (3rd Cir. 1986).

²⁵ From \$770 to about \$15,000, with a suitable reduction for small entities.

²⁶ 35 U.S.C. § 102 (emphasis added).

exploitation by others who don't simply copy it. The protection would be simple to get, inexpensive, and place the applicants' description of the techniques in a repository that would be readily accessible to those wanting to learn and build on the technology - the database of patents and applications maintained online by the patent offices. And that organized prior art would be available to examiners, so that somebody else couldn't get a patent on something unoriginal and force the software developer into expensive litigation to invalidate the patent.