



4-1-2010

Software Innovations and Patent Law Reform

James Reber

Follow this and additional works at: <https://scholarsarchive.byu.edu/byuplr>

BYU ScholarsArchive Citation

Reber, James (2010) "Software Innovations and Patent Law Reform," *Brigham Young University Prelaw Review*: Vol. 24 , Article 28.
Available at: <https://scholarsarchive.byu.edu/byuplr/vol24/iss1/28>

This Article is brought to you for free and open access by the All Journals at BYU ScholarsArchive. It has been accepted for inclusion in Brigham Young University Prelaw Review by an authorized editor of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

SOFTWARE INNOVATIONS AND PATENT LAW REFORM

*by James Reber**

I. INTRODUCTION

To fulfill its constitutional duty,¹ Congress established two separate mechanisms to give exclusive rights of distribution to the creators of useful materials: copyrights and patents. A copyright is a claim on a particular expression of an idea (such as a book about how a machine operates), while a patent is a claim on the idea itself (such as the operation of the machine).² However, the line between copyrights and patents is not always so clearly demarcated.

However, the distinction between patents and copyrights becomes unclear in the software development industry. From the initial widespread use of computers in 1945 to 1995, copyrights were the only formal protection available to software developers. However, a series of court cases expanded patent protection to cover software

* James Reber is a senior majoring in computer engineering. He will attend law school in the fall with an interest in intellectual property law.

1 Article 1, Section 8 of the United States Constitution states: “Congress shall have power. . . [t]o 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.”

2 See UNITED STATES PATENT AND TRADEMARK OFFICE, GENERAL INFORMATION CONCERNING PATENTS (1997), http://www.uspto.gov/web/offices/com/iip/pdf/brochure_03.pdf.

innovations.³ The move toward patents on software culminated in 1995 when the United States Patent and Trademark Office (USPTO) allowed patents on software.

Despite this change, several problems persist in applying patent law to software innovations. The current U.S. patent system discourages third parties from researching and improving patents in two main ways. First, the current system discourages third parties by making would-be researchers of existing patents liable for willful patent infringement.⁴ Second, the patent system fosters uncertainty regarding the scope⁵ of software patents issued by emphasizing the function of a software innovation over the logical structure of the innovation.⁶

3 *See, e.g.*, *Diamond v. Diehr*, 450 U.S. 175 (1981); *In re Alappat*, 33 F.3d 1526 (1994); *In re Lowry*, 32 F.3d 1579 (1994). *See generally* GREGORY STOBBS, *SOFTWARE PATENTS*, § 4.02 (2d ed., Aspen Law and Business 2000).

4 A party is guilty of willful patent infringement when they deliberately copy the patented work of another, completely disregarding the property rights of the second party. A finding of willful patent infringement includes determining the guilty party's intent to infringe and reasonable belief that the patent was invalid, and can have the consequence of multiplication of damages (*Pall Corp. v. Micro Separations*, 66 F.3d 1211, 1221 (Fed.Cir.1995)).

5 The scope of a patent refers to subject matter protected by the patent. The scope is determined by the claims of the patent. *See* UNITED STATES PATENT AND TRADEMARK OFFICE, *A GUIDE TO FILING A UTILITY PATENT* (2008), http://www.uspto.gov/web/offices/com/iip/pdf/brochure_04.pdf.

6 There are larger issues of the patent system that will not be addressed in this article. For instance, a low threshold for obviousness in granting patents (along with broad interpretation of the doctrine of equivalents) allows an early implementation of a software innovation to preempt subsequent disparate implementations of that innovation. In other words, an improved version of an innovation may be found to infringe an earlier (and presumably inferior) patented version of that innovation, even though the two versions are similar only in the results they produce (but are completely different in how they obtain those results).

To remedy these problems, three proposals should be adopted by Congress for patents regarding software⁷ innovations:

- 1) Publish software patent applications prior to examination and solicit public help in collecting applicable prior art⁸ citations.
- 2) Require either written notification of patent infringement from the patent holder or willful copying from a patented source before making a patent infringer liable for willful patent infringement.
- 3) Make software claimable solely in terms of its logical structure and give any awarded patent scope commensurate with the patent's degree of enablement

II. CURRENT PATENT SYSTEM DISCOURAGES THIRD-PARTY INNOVATION

The U.S. patent system requires that inventors provide descriptions of their inventions that are clear and concise enough “to enable [others] . . . to make and use the same.”⁹ However, this does not happen within the software development industry for three reasons: first, the description of the patented innovation is not useful for research because patent claims emphasize the physical nature of the innovation, rather than the logical structure of the innovation (as software engineers understand it); second, any software product inspired by a patented work may unintentionally expose its owner to liability for patent infringement¹⁰ due to the uncertainty caused by describing

7 The term “software innovation” in this article refers to an innovation where the software component is a major inventive step in the innovation; “software patent” refers to patents on software innovations.

8 “Prior art” is the phrase used to describe materials (such as previously-issued patents, articles, or publications) similar to a particular patent application. When multiple articles of prior art are found which in combination overlap all the claims of a particular patent application, the prior art is said to anticipate the patent application, and the patent is denied.

9 35 U.S.C. § 112 (2008).

10 See 35 U.S.C. § 271(c) (2008).

only the physical structure of the original patent; and third, the act of reading a patent may make a software developer liable for willful infringement (and the accompanying triple damages) for an existing software product.

A. Requiring Physicality in a Patent Obscures the Patent's Description

The language of software patents is unsuitable because the patent system has historically required physicality of an invention.¹¹ Before software and business method patents became commonplace in the 1990s, patent protection was mostly given only to those systems which were either physical objects themselves or induced a physical transformation in another object.¹² The written description required for patents describes the nature of the object or the steps of the process necessary to induce physical transformation in the other object. This means that the description outlines what the object is and what it does, not how it performs its operations.

Unlike traditional patented items, software is abstract and intangible.¹³ Yet this standard for the written description still mandates that software be described in terms of a physical structure. Such a description is difficult for software engineers to comprehend. Engineers describe software in terms of its abstract operations and logical elements; merely describing the physical manifestations of a software innovation hides these operations and logical elements and prevents software engineers from productively innovating previous patents.

11 Robert Plotkin, *Computer Programming and the Automation of Invention: A Case for Software Patent Reform*, 7 UCLA J.L. & TECH. 1, 30-34 (2003).

12 *Id.* at 16-17.

13 For instance, common operations in software such as appending two sequences of characters or manipulating processes on a computer are abstract concepts and have no analogue in the physical world. While it is true that such operations in software produce physical changes in the computer, the physical changes are coincidental. The software engineer conceives of the abstract operations, and typically does not care about the physical manifestations of those operations.

B. Current Patent Infringement Doctrine Discourages Using Patents for Research

A second reason the current patent system discourages third-party follow-up is that software products inspired by a patented work expose the developer to liability for willful infringement.¹⁴ When an individual or company is suspected of infringing on the patent of another, the patent holder files suit. If the accused is found to be infringing on the patent, then the accused is fined in proportion to the severity of the infringement. If the infringement is knowingly and willfully done, then the infringer is liable for up to three times the fine levied by the court.¹⁵ These punitive damages discourage malicious competitors from profiting should they deliberately copy the patented product of another.

Unfortunately, these triple damages also have the consequence of discouraging research based on patents. People with an interest in researching patents are also primarily people engaged in those patents' markets, either as competitors or as researchers. These researchers are likely to have existing products similar to patents they research. If these researchers read patents applicable to their market and are later found to have infringed any of those patents with their existing products, they may be liable for willful infringement, even though their products were not intended to infringe on the patents they studied. To avoid this predicament, many would-be researchers do not read patents so that their knowledge of the patent cannot later be turned against them. Desirable third-party innovation is consequently lost due to the fear of being liable for willful infringement.

14 See 35 U.S.C. § 271(c) (2008).

15 35 U.S.C § 284 (2008).

C. Current Patent System Permits Overly-Broad or Non-Novel Patent Scope

Another detrimental effect of the current U.S. patent system is that it unintentionally permits patents on non-novel innovations.¹⁶ Patent examiners spend between 18 to 20 hours on average reviewing each patent application.¹⁷ This is the time for examiners to review and understand the application, find relevant prior art, compare the application to the prior art, and draw conclusions about the application's validity. Some argue that this short amount of time leads to the approval of patents which are anticipated by prior art.¹⁸

III. ATTEMPTS TO IMPROVE THE PATENT SYSTEM

The problems described above can be corrected. Three proposals are now presented. For each proposal, the benefits and flaws of that proposal are analyzed. Remedies for the flaws will be addressed later.

A. Third-Party Review

As stated previously, the hurried patent review process leads to patents of questionable validity, posing threats to the software development industry where innovations occur incrementally.¹⁹ One solution proposed to address this problem is the Peer to Patent program.²⁰

16 See N.Y. LAW SCH. INST. FOR INFO. LAW & POLICY, COMMUNITY PATENT REVIEW PROJECT SUMMARY (2007), http://dotank.nyls.edu/communitypatent/p2p_exec_sum_feb_07.pdf at 2-3.

17 See *id.*; see also FED. TRADE COMM'N, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY, 10 (2003).

18 See N.Y. LAW SCH. INST. FOR INFO. LAW & POLICY *supra* note 17; see FED. TRADE COMM'N, *supra* note 18, at 10.

19 See FED. TRADE COMM'N, *supra* note 18, at 6.

20 Peer to Patent was a collaborative project between New York Law School and the Patent Office. It ran as a pilot program from June 2007 to June 2009. The success of the program is currently under review by both New York Law School and the Patent Office. See generally N.Y. LAW SCH. INST. FOR INFO. LAW & POLICY *supra* note 17.

This program sought public participation in finding, verifying, and applying prior art references to patent applications.²¹ Patent applications were published online and volunteer reviewers examined the published applications over a period of four months. The reviewers submitted prior art references, commented on patent claims or on entire patent applications, and rated prior art submitted by others. At the end of the four month period, the prior art references were forwarded to both the application's patent examiner and the applicant. The examiner then conducted the examination procedure and was free to use as many or as few of the references needed.²²

There are several benefits to using a third-party review system.²³ Patent examiners benefit when third-party reviewers provide both prior art references and commentary to explain those references.²⁴ The general public and the industry related to each individual patent benefit when third-party reviewers provide prior art submissions because these submissions impeach an overly broad or obvious patent. Even the patent applicants benefit; should they receive a patent after their application undergoes third-party review, the applicants have greater confidence that no prior art will be uncovered during litigation which could overturn their patent.

Unfortunately, a third-party review system also has drawbacks. The language used in patent applications is difficult for some software engineers to understand,²⁵ limiting the number of individuals who may participate as reviewers. With the large number of pat-

21 In particular, volunteer reviewers would be sought among those trained in the field of each particular patent, such as university academics and researchers, and industry professionals.

22 N.Y. LAW SCH. INST. FOR INFO. LAW & POLICY, *supra* note 17, at 9.

23 N.Y. LAW SCH. INST. FOR INFO. LAW & POLICY, *supra* note 17, at 3-6.

24 The possibility exists that unskilled reviewers may overload the examiner by supplying numerous irrelevant prior art citations. This is unlikely to happen: Reviewers rank prior art submitted by other reviewers, thus giving higher ranking to the most relevant citation. The examiner is then free to take as many or as few of these references as appropriate. (See N.Y. LAW SCH. INST. FOR INFO. LAW & POLICY, *supra* note 17, at 4.)

25 Plotkin, *supra* note 12, at 35.

ent applications filed each year (456,321 total patent applications in 2008 alone),²⁶ the number of reviewers needed may easily exceed the number of individuals willing to participate. This potential lack of reviewers contributes to a second criticism of the peer-to-patent system: in a contest to impeach each other's patents, large corporations may expend resources that small businesses or individuals cannot match, and so could easily find prior art to cause the small company's patent application to be narrowed or rejected. With its limited resources, a small business would struggle to impeach many of its larger competitors' patents. Thus, while it has many benefits, the Peer to Patent program may allow large corporations to gain an unfair market advantage over smaller competitors.

B. Alter Requirements for Willful Infringement

By making patent researchers liable for willful patent infringement, current United States law discourages third parties from further innovating existing patents. A solution proposed by the Federal Trade Commission would require the patent holder to provide written notice to the alleged offender before they may be accused of willful patent infringement.²⁷ This proposal would change current patent practice by protecting those infringers who did not deliberately copy a patented source.

For instance, consider three parties of patent infringers. The first party consists of those who do not research patents and who market a product which unknowingly infringes on patented materials. Under both the proposal and current patent law, this first party would be liable for patent infringement but not willful infringement. The second party consists of those who may or may not use patents for research and market a product which contains deliberately copied patented materials. Under both the proposal and current patent law, this group would be liable for willful patent infringement. The third

26 UNITED STATES PATENT AND TRADEMARK OFFICE, U.S. PATENT STATISTICS CALENDAR YEARS 1963 – 2008, http://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.pdf.

27 FED. TRADE COMM'N, *supra* note 18, at 18.

party consists of those who conduct research using existing patents and who market a product which still unintentionally infringes on any number of those patents which they researched. Under current patent law,²⁸ this party is liable for willful patent infringement because they found out that their product infringed a patent. Under the proposal, while this group would still be liable for patent infringement, they would not be liable for willful patent infringement unless they were first informed by written notice from the patentee that they were infringing. This provides a window for the party to address their patent infringement. As a result, this proposal removes a major disincentive for third parties conducting research using patents by protecting these researchers.

This written notice proposal does not, however, automatically encourage third parties to research patents. The proposal does not address the difficulty that skilled professionals face in understanding patents. As addressed earlier, the language used to describe software is another obstacle to third-party follow-up innovation. This flaw does not hurt the proposal's feasibility, but does mean that the proposal is an incomplete solution.

C. Make Software Claimable in Terms of Its Logical Structure

Patent claims should describe the structure of software innovations, rather than just describing the behavior of the software.²⁹ For instance, consider U.S. patent 5,883,995, which describes software that limits how many times a user may run a particular computer program:

9. A[n] . . . on-line method for demonstrating software programs to a potential purchaser of the programs, comprising the steps executed by a computer of: receiving from an electronically accessible system a software program to be demonstrated . . .

10. The method of claim 9 wherein the enabling step further comprises the step executed by the computer of

28 35 U.S.C. § 271(c) (2008).

29 See Plotkin, *supra* note 12.

preventing the enabling of the software program when the user has already sampled the software program a predetermined number of times.

These claims explain the behavior that the software performs. Because patent law has historically been applied to physical machines or processes, these claims recite the physical nature of the invention (referred to as the physicality of the invention).³⁰ This recitation must be sufficient to allow one skilled in the applicable field to duplicate the invention.³¹

This description is adequate for many technological fields but describing software this way is more problematic. While describing the behavior of a machine or process is adequate to infer how to construct that machine or process, a description of the behavior of software may refer to multiple ways of achieving that behavior. For instance, in claim 10 of the '995 patent quoted above, the step of preventing execution of the sampled software could occur in any number of ways. First, the potential purchaser's computer may have software which tracks the date and time of each use, and so tracks when the maximum number of uses has been met. Second, the potential purchaser's "computer" may be a video game system which contains an embedded chip with circuitry that opens with each use of the software and so can stop the system when the desired video game has been overused. Third, the purchaser's "computer" may be a cellular phone, the operating system of which tracks the number of times the phone accessed the program's application server, and so it can sever the connection when the maximum number of uses has been met. While all three examples implement the behavior recited in claim 10 of the '995 patent, each example uses a different approach. Nevertheless, each example would be preempted by claim 10.

A solution would be to require software innovations to be claimed in terms of logical structure. For instance, if written in terms of its logical structure, the patent quoted above would recite as follows:

30 See Plotkin, *supra* note 7, at 30-34, for a more detailed analysis of how the physicality requirement came to be and how it affects software patents.

31 See 35 U.S.C. § 112 (2008).

10. The method of claim 9 wherein the enabling step further comprises a list of dates and times when the program is loaded.

11. The method of claim 9 wherein the enabling step further uses the list of dates and times to determine when the user has already sampled the software program a predetermined number of times.

As in this example, claiming software in terms of its logical structure allows software modules with similar behavior, but different ways of achieving that behavior, to still be independently patentable.³²

IV. IMPROVING THE PATENT SYSTEM

Two of the three proposals considered above have flaws. However, if these proposals are taken as a package, the benefits of the other proposals help mitigate the flaws of each proposal considered individually.

The changes proposed herein incorporate the three proposals above: using a system of third-party review to find prior art references for software patent applications, requiring written notification prior to liability for willful patent infringement, and mandating that software inventions be claimed in terms of their logical structure. The benefits of these proposals have already been shown. The effects the complete set of proposals will have on the flaws of two of the proposals are now considered.

32 This is important in computer science, where multiple algorithms exist to address the same problem. For instance, to sort a set of objects (an important step in many computer programs), a programmer may choose to use a *quick sort*, a *merge sort*, a *bubble sort*, a *heap sort*, or an *insertion sort*, to name a few. These different sort algorithms all perform the same operation, but achieve that operation very differently. Under the logical structure proposal, these algorithms would each be separately patentable despite performing the same operation.

A. Implement a Third-Party Review System

The effectiveness of a third-party review system is limited by the language in which software patents are written. The proposal to claim software innovations in terms of their logical structure will make software patents more intelligible to software engineers, and makes the patent applications more accessible to potential reviewers. The problem of having too few reviewers is thus mitigated.

The second problem with a third-party review system is the possibility of parties using their power to impeach their competitors' patents, while their competitors are financially unable to retaliate. This problem may be resolved by publishing the patent anonymously during the period when public review is permitted. While this solution does not prevent a party from indiscriminately attacking all patents they feel encroach on their markets, it does prevent that party from deliberately attacking a single competitor, stopping a given party from using the review process to undermine their competitors.

B. Alter the Requirements for Willful Patent Infringement

As described previously, the proposal by the Federal Trade Commission to alter the requirements for willful patent infringement does not by itself allow researchers to easily read software patents for research purposes. Describing a software invention in terms of the physical nature of the invention obfuscates how the software actually operates, degrading its quality as a research tool. Describing software in terms of its logical structure (instead of its physical structure) is more common among software professionals. A description of the software's logical structure would provide the software professional with a more complete description of how the software operates and would thus more completely fulfill 15 U.S.C. § 112: "The specification shall contain a written description of the invention . . . to enable any person skilled in the art to which it pertains . . . to make and use the same. . ." Including the logical structures proposal will help bring about the effect the willful infringement proposal seeks.

V. CONCLUSION

Computer software continues to change the way humans interact. Software has exploded in complexity and diversity since the early days of computer in the 1950s. But the nuances and nature of software have caused inconsistent treatment of software by the law. This is evidenced by the variety of court decisions regarding software patents.

The proper treatment of software innovations by patent law will continue to be debated. In the mean time, there are changes that can be made to help the United States patent system better fulfill its purpose “[t]o promote the progress of science and useful arts . . .”³³ While the package of changes proposed herein do not resolve all weaknesses of the patent system, it nonetheless helps the patent system produce higher quality patents on software innovations and better encourage software innovation.