ENHANCING THE QUALITY OF SOFTWARE PATENTS BY OPEN REVIEW

Lung-Sheng Chen[*]
Assistant Professor of Law
Department of Financial & Economic Law,
National Chung Cheng University (Taiwan)

ABSTRACT

Software is an abstract technology that facilitates abstract claiming. In order for the patent system to operate well, it is crucial that the patent examiners secure the quality of software patents. The examiners, however, have difficulties in obtaining “good quality” prior art information regarding a particular patent application. Due to the institutional factors, the examiners are not willing to seek outside resources. As such, the patent review practice cannot work well due to the information deficit problem that consequently leads to the poor patent quality problem. In particular, as this article noted, the poor quality of software patents has posed serious threats on the open source community. With this concern in mind, this paper discusses the ways to mitigate the risk of patent infringement under the open source framework, such as the GNU GPL version 3.0. This article traces the open source community’s patent problem back to the patent review process’s poor quality problem. This article, therefore, explores different ways to improve the quality of software patents, suggesting the U.S. patent examination system to include open review into its existing review framework. On the one hand, the traditional peer review will provide valuable insights for the examiners. On the other hand, however, a new review model that invites the public to contribute their knowledge and to scrutinize the review process is of great help to enhance the quality of an issued patent, especially a software patent.

Keywords: Patent quality, software patent, community patent review, pilot program, open source

[*] J.D. 10’ & LL.M. 08’, Washington University in St. Louis School of Law; LL.M. 01’, College of Law, National Chung Cheng University, Taiwan; B.A. in Law 97', College of Law, National Chengchi University.

I. Introduction

"Sharing… is cool."[1] We are constantly sharing something, online or offline, with our family, friends, colleagues, neighbors, or persons we even don't know at all. A sharing paradigm can be found in the computing community—the open source movement. In 1983, Richard M. Stallman announced the GNU project that consequently launched the open source movement.[2] This project developed a free computer operating system open to computer users. Each user can download and use the GNU's software by agreeing the GNU General Public License ("GPL"),[3] under which the user, as a licensee, is free to use, share, and change the licensed software, as well as to share the changes that the user makes based on the software.[4] In addition, the user can access, modify, and distribute the software's source code and object code.[5] Via the GPL and the license agreements of this sort, the open source movement has developed a sharing paradigm that values the user's freedom.

While sharing is so cool, some people just don't think so, however. Many open source individuals, groups, or firms that grant license to users have faced threats of their competitors' patent infringement claims. Facing this "patent problem," some open source members have developed a strategy to crash their competitors' efforts in obtaining software patents. That is, these open source members are quite often injecting "new" information or discoveries into the public domain; as such, their potential competitors' patent infringement claims is likely to fail due to the lack of the novelty requirement.[6] Professor Robert P. Merges calls the practice of this sort "Property-Preempting Investments."[7] In addition to the PPIs, the open source members address the "patent problem" by including into their license agreement a "patent clause." For instance, the GNU GPL in its version 3.0

[1] Neil M. Richards, *The Peril of Social Reading*, 101 GEO. L.J. (forthcoming 2013), *available at* SSRN: http://ssrn.com/abstract=2031307.

[2] For an introduction of this project, *see* GNU Operating System, Overview of the GNU System, http://www.gnu.org/gnu/gnu-history.en.html (last visited May 23, 2012).

[3] Brian Carver, Note, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 BERKELEY TECH. L.J. 443, 444 (2005).

[4] Brett Smith, *A quick guide to GPL v3*, *available at*: http://www.gnu.org/licenses/quick-guide-gplv3.html (last visited2012/5/21).

[5] GNU GPL version 3.0, Preamble, *available at*: http://www.gnu.org/licenses/gpl-3.0.html (last visited May 21, 2012); Carver, *supra* note 3, at 456.

[6] Carver, *supra* note 3, at 461; Sara Boettinger & Dan L. Burk, *Open source patenting*, 1 JIBL 221, *available at*: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=64512 (last visited May 23, 2012); Robert P. Merges, *A New Dynamism in the Public Domain*, 71 U. CHI. L. REV. 183, 185 (2004).

[7] Merges, *id.*, at 185; Carver, *supra* note 3, at 462-463.

provides its users (licensees) with an explicit patent license protection against the patent claims by this project's contributors and redistributors.[8] As Stallman puts it, "the only way to make software movement safe is to abolish software patents."[9] Nonetheless, as this article will show, the GPL's patent clause might be insufficient to address the open source software's patent problem. Tracing back this patent problem to an earlier stage, this article argues that to resolve the patent problem requires a reform of the patent examination's review process.

Software is an abstract technology that facilitates abstract claiming. The job for patent examiners to secure the quality of software patents becomes crucial. However, patent examiners are quite restricted to accessing the information of the prior art of a particular patent application. The current review practice's reluctance to use outside science and isolation of expertise causes an information deficit problem and consequently results in poor quality patents.[10] This paper discusses ways to improve the quality of issued software patents by reforming the patent examination system. This paper argues that such reform is compelling not only because it will improve the U.S. patent system itself, but also because it will advantage the open source community by reducing its patent problem.

Part II describes the insufficiency protection against patent infringement under the GNU GPL version 3.0. Part III illustrates the information deficit problem as well as the quality problem in the current patent system. Finally, part IV outlines detailed reform proposals including tradition peer review and the USPTO Community Patent Review Project and discusses some potential problems. This paper concludes that the patent examination system should reform to enable the examiners to access outside input. Open review, (such as traditional peer review and the USPTO Community Patent Review Project) builds a promising framework that the public can participate in the examination process of a particular patent review process and can better resolve the information deficit problem and promote the patent quality.

II. The GPL's Patent Clause and Its Insufficiency

GNU GPL version 3.0 attempted to deal with the software patent threats and provide users with an explicit patent license covering any patents held by a developer of a GPL-covered work. The relevant patent rights license clauses are set forth in the Preamble and Section 11 of the GNU GPL version

[8] Richard Stallman, *Why Upgrade to GPL v3, available at* http://www.gnu.org/licenses/rms-why-gplv3.html (last visited May 23, 2012).

[9] *Id.*

[10] Beth Simone Noveck, *Peer to Patent: Collective Intelligence, Open Review and Patent Reform*, 20 HARV. J.L. & TECH. 123, 132 (2006).

3.0.

A. The Preamble

The Preamble of GNU GPL version 3.0 informs the users of the patent problem. It provides:

> Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do; we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GNU GPL version 3.0 assures that patents cannot be used to render the program non-free.[11]

The GPL thus requires those who intend to distribute software under the GPL framework to permit free redistribution of their contribution to the program. Since the GNU GPL version 3.0 adds a new section 11, as discussed below, the Preamble of the GNU GPL 3.0 contains some revision accordingly; nonetheless, the main idea of the Preamble remains as its previous version—the GNU GPL version 2.0.[12]

B. Section 11: Patent Grant Clause

As amended, Section 11 of the GNU GPL version 3.0 expressly provides users with a patent grant. This new Section provides that a distributor of a GPL licensed work automatically grants a non-exclusive, worldwide, royalty-free patent license of his essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the software.[13] Under this provision, a distributer cannot make any patent infringement claims against a redistributer who uses the distributor's contribution.

Section 11 attempts to make the subsequent use of a GPL-covered work free from any patent infringement claims. If a distributor has actual knowledge that the conveyance or use of the GPL-covered work would

[11] GNU GPL version 3.0, Preamble, *supra* note 5.

[12] The Preamble of GNU GPL version 2.0 provides that "Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all. The precise terms and conditions for copying, distribution and modification follow. *See* GNU General Public License v2.0, Preamble, *available at*: http://www.gnu.org/licenses/gpl-2.0.html (last visited May 23, 2012).

[13] GNU GPL version 3.0, *supra* note 5, section 11.

infringe one or more identifiable patents, the redistributor should make the corresponding source of the work available for anyone to copy, free of charge, deprive him of the patent license, or extend the patent license to downstream recipients.[14] The aim of the provision is to protect the downstream users from patent infringement allegations made by upstream distributors.

There are concerns over the patent grant clause of GNU GPL version 3.0, however. First, although the patent rights grant provision broadens the freedom of distribution of software, it runs counter to the traditional logic and goal behind the patent system. GNU GPL version 3.0 requires a distributor not to insist on the patent right he may hold in the GPL-covered work. The patentee thus waives his legal rights against not only the parties to whom he distributes the GPL-covered work but also everyone in the GPL framework. This is true even after a distribution has withdrawn from the GPL framework. As such, many companies refuse to apply GNU GPL version 3.0 because they worry that this provision will could force them to abandon their patent rights against everyone involved in the GPL framework.[15]

Additionally, GNU GPL version 3.0 cannot completely mitigate the risk of patent infringement. For instance, a program developer writes software programs and is unaware of the fact that his work would fall into another existing patent claim. Under this circumstance, the program developer and the consequent recipients will still face the risk of being sued for infringing patent by the patent holder. No solutions for this circumstance can be found in the GNU GPL version 3.0. It then seems that the only way to mitigate the risk of infringement litigations is to improve the patent examination procedure to enhance the quality of software patents.

III. Information Deficit and the Quality Problem of Software Patents

[14] *Id.* Section 11 of GNU GPL version 3.0 provides that "if you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid."

[15] Shaobin Zhu, *Patent Rights Under FOSS Licensing Schemes*, 4 SHIDLER J.L. COM. & TECH. 4 (2007).

Because a software patent claim often contains abstract language, software patents are often subject to litigations or disputes over the scope of a particular software claim.[16] The abstract nature of software technology makes it more difficult to place limits on abstract claims in software patents.

A. Patent Quality

The patent system serves as a mechanism to encourage innovations by granting inventors an exclusive right within a limited period of time. However, several issued patents have been questioned as they claim either unpatentable subject matters or are obvious or not novel.[17] These "bad" patents decrease the quality of the patent system and are often abused by patent trolls who obtain patents not for manufacturing products.[18]

A patent's quality will be measured before and after the patent is issued. First, during the patent application review process, the statutory patentable requirements, such as requirements regarding a patentable subject matter,[19] novelty,[20] non-obviousness[21] and utility[22] serve as one way to measure the patent quality.[23] In order to assess the quantity of patent quality, the USPTO gives most weigh on its internal quality assessment audits.[24] Second, a patent's quality will be assessed in subsequent legal proceedings after that patent being issued.[25] The rate that patents are invalidated in subsequent proceedings reflects the quality of all issued patents.[26]

Reviewing and construing the prior art of a particular patent invention are keys to a patent invalidation proceeding.[27] The invalidation of an issued patent indicates that the prior art is not being thoroughly searched and discovered during the review process of that patent. Therefore, identifying prior art becomes crucial to improve patent quality.

[16] MICHAEL MEURER & JAMES BESSEN, PATENT FAILURE 9-11 (2008), *available at*: http://researchoninnovation.org/dopatentswork/dopat9.pdf (last visited May 23, 2012).

[17] Christopher J. Worrel, *Improving the Patent System: Community Sourcing and Pre-Grant Opposiion*, 42 U. TOL. L. REV. 833, 835 (2011).

[18] Susan Walmsley Graf, *Improving Patent Quality Through Identification of Relevant Prior Art: Approaches to Increase Information Flow to the Patent Office*, 11 LEWIS & CLARK L. REV. 495, 496-99 (2007) (discussing the patent quality and problems with patent trolls).

[19] 35 U.S.C. § 101 (1952).

[20] 35 U.S.C. § 102 (2011).

[21] 35 U.S.C. § 103 (2011).

[22] 35 U.S.C. § 101 (1952).

[23] Graf, *supra* note 18, at 499.

[24] *Id*. at 500-01.

[25] Bronwyn H. Hall & Dietmar Harhoff, *Post-Grant Reviews in the U.S. Patent System - Design Choices and Expected Impact*, 19 BERKELEY TECH. L.J. 989, 991 (2004).

[26] *See* Graf, *supra* note 18, at 501-02.

[27] *Id.* at 501.

B. Problems in Identifying Software Patents Prior Art

Commentators have noted that many software patents contain broad and vague claims; however, a close examination of the claims of these sort indicate that such software contain very limited "invention."[28] As result, when examining a software patent application, to do a diligent prior art search is crucial to the software patents' quality problem.

At the core of the software patent quality problem is information access. Due to the patent examiners' lack of access to sufficient and adequate information, many commentators consider that the USPTO is awarding software patents improvidently.[29] In particular, some allege that the patent examiners are not quite experienced with the patent examination. Examiners are supposed to possess "the knowledge that comes from specialized experience," but fifty-five percent of patent examiners have just been employed by the USPTO for less than two years. Moreover, they are not required to have an advanced degree.[30] This reflects that over half of the examiners are not qualified to be expert bureaucrats.

In addition, the examiners' resources for prior art search are limited. When examining a patent application, patent examiners rely on three computer databases systems[31] to search and access the prior U.S. patents, foreign patent abstracts, certain pending U.S. applications, and additional proprietary database libraries.[32] Moreover, the examiners are restricted to use Internet as a search tool for security reasons.[33] In other words, the examiners are not independently finding the prior art and are limited to internal sources available at the USPTO when doing the examination.

Further, third parties are restricted to actively provide information. This is because Congress has required the USPTO to "establish appropriate procedures to ensure that no protest or other form of pre-issuance opposition to the grant of a patent on an application may be initiated after publication of the application without the express written consent of the applicant"[34] A third party must provide his input by mail within two-month window, plus a

[28] MEURER & BESSEN, *supra* note 16, at 9-16.

[29] Noveck, *supra* note 10, at 132.

[30] *Id.*

[31] The three databases at USPTO are Examiner's Automated Search Tool ("EAST"), Web-Based Examiner Search Tool ("WEST"), and Foreign Patent Access System ("FPAS"). *See* United States Patent And Trademark Office, MANUAL OF PATENT EXAMINING PROCEDURES (hereinafter "MPEP") §902.03(e) (8th ed. 2001) (latest revision July 2010).

[32] *See id.*

[33] MPEP, *supra* note 31, §904.02(c); Noveck, *supra* note 11, at 135.

[34] 35 U.S.C. §122(c) (2011).

fee and no commentary.[35] Thus, there are fewer third party input available. The burden of finding adequate information for patent examination greatly falls to the patent examiners.

IV. Proposals

One of the thoughtful solutions to the information deficit and quality problems is Beth Simone Noveck's *Peer to Patent: Collective Intelligence, Open Review and Patent Reform,*[36] which abandons the traditional peer review model and provides a new reform model for open patent examination (Peer-to-Patent). While Noveck's Peer-to-Patent proposal is insightful, this article contends that traditional peer review still works well and should co-exist with the Peer-to-Patent model.

A. Traditional Peer Review

Noveck argues that traditional peer review is inappropriate to solve the information deficit problem because of its lack of transparency.[37] She considers peer review as a fairly conservative means to address the information quality problem. To the extent that she proposes to abandon the traditional peer review, this article disagrees with her arguments for the following reasons.

First, it is adequate to select certain kinds of industry and academic experts to form peer review groups because patent applications, especially new technology such software and biotechnology, are highly professional and technical. Instead, to invite the public to join the peer review group without any qualification requirement will apparently cause inefficiency for gathering unrelated and inadequate information. It is necessary for an experienced and knowledgeable expert to contribute valuable information. And this is just how the traditional peer review works. Noveck argues that the peer review groups are selected based on educational or social status, and this will exclude otherwise qualified and meaningful contributors.[38] However, this is just an administrative and management issue. A solution to this problem is not to repudiate the traditional peer review. Rather, it can be resolved by broadening the list of peer review group, or renew it frequently.

Second, Noveck argues that the peer review member selection process is closed especially when the subject matter relates to high technological issues. In the circumstance of this sort, only certain experts will be invited to join

[35] *See* MPEP, *supra* note 31, at § 1134.01; 37 C.F.R. § 1.99 (2006).

[36] Noveck, *supra* note 10, at 139.

[37] *Id*. at 138.

[38] *Id*. at 141.

the review group.[39] She thus contains that the traditional peer review selection process is not transparent. This article agrees with her argument that the peer review should open to the public; however, if the subject matter concerns certain highly technological knowledge or professional issues, whether the selection process is open or not does not matter a lot. In this circumstance, a selection process that merely opens to the experts belonged to the field will improve the efficacy of the review process.

The traditional peer review in many aspects works well to resolve the examiners' information deficit. Contribution from the industry experts and academic professionals provides the examiners with many valuable insights that improve the quality of the review process. This article thus contends that the traditional peer review can coexist with a new review model, as explored below.

B. USPTO Community Patent Review Project

To encourage the knowledgeable public to submit information regarding patent applications' prior art for the examiners' consideration, the USPTO executed a Community Patent Review Project (hereinafter "CPRP") since June 2007.[40] Adopting Noveck's open review proposal, the CPRP puts the idea of open review into practice by building a framework that separates technical issues from legal disputes.[41] The project "enabled the public to submit prior art and commentary relevant to the claims of pending patent applications."[42]

In light of the information deficit and the examiners' excessive caseload, the CPRP reviewed inventions regarding computer technology, software, information security, or business methods, E-Commerce, management or cost/price determination data processing.[43] Merely training the patent examiners in computer technology does little help in maintaining the patent

[39] *Id.*

[40] The USPTO Community Patent Review Project (hereinafter "CPRP"), which is also known as Peer-to Patent: Community patent Review pilot program, is executed by the "Institute for Information and Policy" of New York law school through the peer-to-patent website at http://www.peertopatent.org. *See* About Community Patent, N.Y. Law Sch., *available at* http://dotank.nyls.edu/communitypatent/about.html (last visited May 21, 2012).

[41] Noveck, *supra* note 10, at 127. Noveck's Peer to Patent model separates scientific from legal decision making by means of an online network that the scientific community provides what it knows best—scientific information relevant to determining the requirement of a patent application.

[42] CPRP, *supra* note 40.

[43] *Id. See also* CENTER OF PATENT INNOVATIONS AT N.Y. LAW SCHOOL, PEER TO PATENT SECOND ANNIVERSARY REPORT 14 (2009), http://dotank.nyls.edu/communitypatent/CPI_P2P_YearTwo_lo.pdf [hereinafter "the 2009 Report"].

quality because the examiners rely mostly on the U.S. database in searching the prior art.[44] While the development of new technology in the field of software is quite rapid, patented inventions within this filed were not available until recently. As such, when an examiner is searching software prior art, he/she could not merely rely on the patent database; instead, the non-patent sources should be given equal, or much more, weight.[45]

The CPRP broadens the current rules to permit third parties—the Peer to Patent community—to submit prior art with their commentary online.[46] The CPRP creates a platform that allows third parties to rank the prior art for relevance, creating a searchable and sorted list of information for the USPTO. Although the CPRP forwards only that "top ten" list to the examiner,[47] the examiners still have access to the full list by searching online database. The USPTO will not interfere with the process of CPRP since each examiner is independent of the USPTO. In each patent application review process, the patent examiner remains the ultimate decision power. The project has finished its second year program in 2009, and an examination of this project concluded, like that of the first year program, the CPRP successfully invited public consultation that helped improve the patent quality.[48]

As an open patent review model, the CPRP supplies rich information revealing patent applications' prior art by encouraging third parties to participate in the patent examination process. Nonetheless, the open review model is not without its own difficulties and problems.

The first problem is that a contributor who provides the CPRP with certain copyrighted prior art information might face liability for copyright infringement. When a third party uploads a copyrighted article, computer program code, or other information, the uploading might constitute a unauthorized copying which results in copyright infringement for the contributor as well as secondary liability for those administering the CPRP.[49] Although in this circumstance, the contributor might argue that his/her uploading is a fair use,[50] potential litigations of this sort might substantially harm the project's efficacy. A solution to this copyright problem is to give

[44] James Gleick, Patently Absurd, N.Y. TIMES MAG., Mar. 12, 2000, § 6, *available at* http://www.nytimes.com/library/magazine/home/20000312mag-patents.html (last visited May 21, 2012).

[45] Robert P. Merges, *As Many As Six Impossible Patents Before Breakfast: Property Rights for Business Concepts and Patent System Reform*, 14 BERKELEY TECH. L.J. 577, 589 (1999).

[46] The website is http://www.peertopatent.org/ (last visited May 21, 2012).

[47] Noveck, *supra* note 10, at 148-49.

[48] *See* the 2009 Report, *supra* note 43, at 11.

[49] Noveck, *supra* note 10, at 146.

[50] *Id.* at 147.

notices on the Peer-to-Patent website that if contributors are submitting a copyrighted work to the project, they must quote the materials and/or indicate the source of the submitted materials.

Second, the project might attract the competitors of a certain patent applicant to participate in the review process.[51] On the one hand, the project cannot work well without the participation of various contributors, including the applicant's competitors, who desire to disclose a "good quality" prior art with intent to defeat the patent application. On the other hand, however, it is likely that some contributors, especially the competitors, might cheat the ranking system, misleading the examiners by giving untrue information. Regarding this point, Noveck suggests that the system "must control against participants voting 'early and often.'"[52] In addition, a transparent framework that receives the public's scrutiny and a clear policy that expressly forbids the gaming practice are required to prevent this problem.[53]

Moreover, the prior art information provided by the contributors is of great help for examiners to determine a particular application's novelty. Nonetheless, when an examiner reviews an application's non-obviousness, the prior art information contributed from the open review might not be sufficient. This is because to observe the non-obviousness requirement, an inventor must have some "advances" in his/her invention. To determine whether an invention is non-obvious, the examiners should not rely too much on the prior art information provided by the contributors.

V. Conclusion

The free software movement is facing serious patent problems. Although the GNU GPL v. 3.0 contains a broad patent grant provision that extends protections for its users, it is still insufficient for the users to avoid the risk of patent infringement. While the patent system provides an incentive for firms and individuals to invest in new technologies, there is an important basic question: how to ensure and maintain the patent quality. To address this question, this article suggests a reform of the patent examination system, in particular, the review process of software patent applications.

Open review presents a practical and efficient alternative by enhancing the institutional competence of the patent examiners. At the core of the software patents' quality problem is the examiners' information deficit problem. To this point, the traditional peer review could provide valuable information regarding technological knowledge and professional information by technology and academic expert. On the other hand, a new review model,

[51] *Id.* at 149.

[52] *Id.*

[53] *Id.*

the CPRP, offer an opportunity to welcome outside contributions. Both reform proposals should co-exist under the current patent system to spur empirical and data-driven reform[54] and to supplement, not replace, substantive examination by the examiners. After all, without effectively addressing the patent quality problem, a patent system is unlikely to work best and achieve its purpose to promote the industry and the progress of science and the useful art.

Cited as:
Bluebook Style: Lung-Sheng Chen, *Enhancing the Quality of Software Patents by Open Review*, 1 NTUT J. OF INTELL. PROP. L. & MGMT. 41 (2012).

APA Style: Chen, L.-S. (2012). Enhancing the quality of software patents by open review. *NTUT Journal of Intellectual Property Law & Management*, *1*(1), 41-52.

[54] *Id.* at 161.