

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Request for Comments
Regarding Prior Art Resources for Use
in the Examination of Software-Related
Patent Applications

Docket No. PTO-P-2013-0064
79 Fed. Reg. 644

**COMMENTS OF PUBLIC KNOWLEDGE,
THE ELECTRONIC FRONTIER FOUNDATION,
AND ENGINE ADVOCACY**

Attn: Seema Rao, Director, TC 2100
Mail Stop Comments—Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
SoftwareRoundtable2013@uspto.gov

Public Knowledge, the Electronic Frontier Foundation, and Engine Advocacy respectfully submit the following comments in response to the Request for Comments Regarding Prior Art Resources for Use in the Examination of Software-Related Patent Applications dated January 6, 2014.

In brief, we recommend that the Patent and Trademark Office engage with communities of small software developers, including the startup and open source¹ software communities, to develop prior art resources. These groups produce much of the most advanced software technology today, meaning that they are among the most valuable sources of prior art in the software field. But they often lack the resources of large companies, so they are less likely to file patent applications or otherwise generate prior art in ways traditionally expected by the PTO. Accordingly, successfully harnessing this body of prior art will require collaboration between the PTO and these communities.

¹ Throughout these comments, the term “open source” is used to refer to the software projects variously termed “free,” “open source,” “libre,” and the like. For further reference see, for example, Terry Hancock, *The Jargon of Freedom: 60 Words and Phrases with Context*, Free Software Magazine, July 24, 2010, available at <http://fsmsh.com/3360>.

I. Introduction: A Personal Account

To begin, the author of these comments would like to relate a personal experience on prior art searching, to highlight both the value of open source software prior art and the issues in using it.

In 2004, having just graduated college with a degree in Computer Science, I was hired as a consultant for a patent lawsuit. My job was to find prior art that could be used in the case, which related to networking technology.

When I was first briefed on the subject matter of the patents in that case, I immediately recognized that the technology was very similar to an open source computer program. I had used that program myself, and knew that it had been available for years before then. It was clear that this program was highly relevant prior art.

However, there was a significant challenge in proving that the program had the necessary features at the critical date of the patent. It was easy to show that the program existed as of the critical date, and easy to prove that the latest version of the program had all the relevant features. But the program had gone through numerous revisions over the years, with various features being added at different times. All of the information was available—because the program was open source, all of the historical source code was timestamped, archived, and publicly available—but that information was disorganized, understandable only after a great deal of effort.

I searched through archived versions of the website of the program's author. I read through records of changes to the program in the source code repositories. I even reviewed and tested the historical source code of the program itself. Finally, after several days of analysis, I could finally determine that the necessary features were present a year prior to the priority date of the patent. The materials I ultimately presented were simple, comprising just a few archived manuals and documents, but locating and identifying those materials was difficult to say the least.

II. The Open Source and Startup Software Communities Are Highly Innovative Groups Who Generate Cutting-Edge Prior Art That Is Often Overlooked

There are two lessons to be drawn from this story. First, open source software, like much software being produced by small developers, is an incredibly important resource for prior art searching. The communities producing these software programs are large, and the software they produce is on the cutting edge of the computer software arts. Furthermore, open source software is a resource particularly amenable for use as § 102(a)(1) prior art, because all parts of the program are freely available for public inspection.

But second, this trove of prior art is not easy to use in its current form. Source code repositories are designed for software developers, not patent examiners. Lawyers and researchers, like the author of these comments, have the time to focus on a single case and review the evidence in detail. But patent examiners do not have days or weeks to spend on research. Open source software, startup software, and other software will be usable by examiners only if records of that software are marshaled into an appropriate form.

Traditionally, the PTO has relied on innovators to file patent applications in order to build a library of prior art. But startups and open source developers often do not file patent applications, due to lack of funds, desire for flexibility in product design, importance of minimizing time to market, uncertainty about the value of patents, and other reasons. As a result, the body of prior art generated by these innovative software developers is often overlooked during a prior art search. Separate solutions, particularly developed for this type of prior art, are required.

Furthermore, the open source model of software development is conducive to the generation of valuable prior art, but not conducive to organizing that art in a searchable form. Open source software is built by distributed community effort, with a variety of companies, organizations, and individuals contributing to parts of a work. Thus, documents and source code may be distributed across various message boards, websites, servers, and other

locations.² To build useful prior art resources, the open source community will require the PTO's knowledge of what information is relevant; the PTO will require the open source community's knowledge of where that information is stored.

Building useful prior art resources will thus require a joint collaborative effort between the PTO and outside stakeholders, particularly the software creators and developers in the open source and startup communities. Accordingly, we urge the PTO to engage with these communities of developers to develop appropriate resources for prior art searching.

III. The PTO Should Work with the Open Source and Startup Communities, on Strategies for Organizing Prior Art Resources

In order to take advantage of the body of prior art being generated by the open source and startup communities, the PTO should engage in an effort to develop effective, public, searchable prior art resources. This effort will be most effective through collaboration between both the PTO and members of those communities. The following are several benefits that may come out of such collaboration.

A searchable database of software programs. Effective examiner searching requires a text-searchable, date-searchable database. The information necessary to build such a database is already available: open source software is often stored in timestamped code repositories, and startup products are often described in dated press releases or other media venues. Indeed, code repository services such as GitHub and SourceForge already maintain substantial volumes of software data, but that data is not searchable by date at the moment. Thus, through appropriate collaboration, the PTO could without great difficulty build a valuable prior art search database from this raw material.

² See Brief of Amici Curiae Electronic Frontier Foundation, Public Knowledge, Computer & Communications Industry Association, and Apache Software Foundation in Support of Petitioner 6–7, *Microsoft Corp. v. i4i Ltd. P'ship*, 131 S. Ct. 2338 (Feb. 1, 2011) (No. 10-290), available at https://www.eff.org/files/232537_pet_center.pdf.

Seeing past obfuscation to find the most relevant prior art. A common concern about software patents is that applicants are able to use invented terminology to avoid prior art. For example, rather than use a common term of art (such as ‘resolution’) an applicant might deliberately select a more obscure term (such as ‘pixel density’).³ Some of these problems are inherent in using language to explain software, while others are exacerbated by certain applicants’ efforts to expand claim scope.⁴ Examiners faced with an applicant’s invented language will have more difficulty finding the most relevant prior art. To solve this problem, patent examiners should 1) insist applicants use either well-defined or commonly understood terms,⁵ and 2) look beyond any deliberately obscure language to conduct prior art searches tied to the functionality claimed by the applicant.

By working with software developers who are actively engaged in the field, examiners can learn to recognize this deliberate obscurity. This knowledge will assist them in examining applications more accurately. Accordingly, interaction between small software developers and the PTO will not only improve prior art searching, but patent quality overall.

Interaction between examiners and software developers. The PTO benefits already from companies that offer training sessions and presentations to examiners. From these, examiners learn about the latest developments in technologies as well as background information on fields of art.

³ See Joel Spolsky, *Victory Lap for Ask Patents*, July 22, 2013, <http://www.joelonsoftware.com/items/2013/07/22.html> (explaining that it took only 15 minutes to find invalidating prior art for a Microsoft patent application that “used terms like ‘pixel density’ for something that every other programmer in the world would call ‘resolution’”).

⁴ Fed. Trade Comm’n, *The Evolving IP Marketplace: Aligning Patent Notice and Remedies with Competition* 83, 85 (2011), available at <http://www.ftc.gov/os/2011/03/110307patentreport.pdf>.

⁵ We explain this in more detail in earlier comments on the use of glossaries to improve clarity of claim terms. See Comments of Public Knowledge and the Electronic Frontier Foundation, *Strategies for Improving Claim Clarity*, October 24, 2013, <http://www.publicknowledge.org/files/comments-pto-roundtable.pdf>.

Obviously, the PTO would similarly benefit from learning from open source software developers and startup entrepreneurs. But a large company can allocate several of its employees for a training session with the PTO; a small developer or company often has neither the time nor the resources to do so. Thus, to gain the unique perspective of these groups, the PTO will need to seek out opportunities to learn about developments in the small software developer world. For example, startup entrepreneurs often participate in “demo days” or presentations directed to investors; examiners could attend these sorts of events. Indeed, even increased socialization between examiners and members of the open source and startup communities would be helpful.

IV. Conclusion

For the foregoing reasons, we respectfully recommend that the PTO work with the startup and open source software communities to build prior art resources. Doing so will expand the universe of prior art available to examiners, increase the institutional knowledge available to the PTO, and improve the quality of patents.

The commenters would welcome the opportunity to discuss these and other ideas further, and to work with the PTO to bring about this beneficial collaboration with the small software developer communities.

Respectfully submitted,

Charles Duan
Director, Patent Reform Project
Public Knowledge
1818 N Street NW, Suite 410
Washington, DC 20036
(202) 861-0020
cduan@publicknowledge.org