

Kickstarting an Old Patent System for the New Software Era

DANIEL R. BROWNSTONE

Fenwick
FENWICK & WEST LLP

Software patents have been having a rough time of it lately. It seems everyone has something bad to say about them, from the courts to the press, and even some software engineers.

And truly it is software that is being singled out as the villain in the patent universe—Vanity Fair has yet to publish an expose on “the razor blade wars,” nor has the Supreme Court addressed the important question of whether a chainsaw is merely an abstract idea. Patents on starting a car with just one click have seldom been the subject of disdain in the blogosphere.

This nadir, however, presents an opportune time to look for creative solutions to patent protection for software, including a new opt-in alternative for obtaining software patents.

Fundamentally, software is different from other kinds of technologies. Unlike mechanical devices we can see, touch and observe directly in operation, software is invisible, hiding out inside our television sets, smartphones, refrigerators and cars.

While in theory a software innovation should be entitled under the existing patent statutes to the same protection as a new physical machine, the courts and the public often act as though patents on software are allowing applicants to get away with something undeserved. A new kind of windshield wiper blade is clearly worth protecting—we can literally see the improvement. But the software application used to design the blade to have such a high degree of effectiveness? Well, that’s just creating some algorithms—maybe some hard math, but nothing patent worthy.

1. Creating a New Software Patent System

Many in the patent bar and on the courts have struggled mightily for over two decades to articulate the rationale by which existing law should cover innovations in the software industry just as robustly as in the physical world. While those arguments remain valid, there is something to be said for treating software differently for purposes of patent protection. Not because it is less

deserving, but because our system for granting and enforcing patents is out of sync with how the industry operates.

The software industry differs from other industries in several respects, and many of those inform how we might design a new software patent system. Some things to consider are speed, enforceability and term.

Consider the pace of software evolution, in which applications are often refreshed, revised and upgraded within periods of months or just a couple of years. Now consider that under the existing patent system, it can easily take three to five years to get an issued patent that can be enforced against an infringer. A new software patent system, therefore, must result in faster issuance of patent rights through a modified examination system. Under the current system, examiners study each claim of each application for both novelty and obviousness. Under a modified system, examiners could simply search for novelty, leaving the question of obviousness to litigation as is now done in New Zealand. Alternatively, software patent applications can be limited to a small number of claims, or the applicant could designate which claims should actually be examined. Software patents would then issue much more quickly, allowing them to be enforced while the innovations they describe are still technologically relevant.

Next, we can adjust the patent term. Today’s utility patents have a term of 20 years measured from their earliest non-provisional filing date. Adjusting for various examination delays and statutory extensions, patents are typically enforceable for about 15 to 17 years. In many industries, patents grow more valuable with time. This is perhaps most true in the pharmaceuticals industry—consider the growth of Viagra and Lipitor sales over their patent term. In those industries, a longer term is key.

Not typically so for software. Innovations in this space typically have limited longevity. One of the standard battle cries against today’s software patents is they end up being asserted years later against unsuspecting defendants through tortured interpretations of their

claims that go well beyond the scope of what the inventor originally envisioned. A shorter term for software patents—seven years, perhaps—would address this concern while also mitigating the anticompetitive effect inherent in patents.

The third aspect of a new software patent scheme relates to enforceability. In the existing patent system, all issued patents are presumed by statute to be valid. The onus is on a defendant in litigation to prove otherwise by clear and convincing evidence. A defendant who can prove the patent is invalid escapes paying damages for infringement, but nonetheless is left with a huge legal bill for his trouble.

2. Shifting the Litigation Burden

In a modified software patent system, the tradeoff for a less rigorous examination to obtain the patent is an increased burden on the patent owner in litigation. This may include, for example, a lower standard imposed on the defendant to invalidate—for example, a preponderance of the evidence standard instead of a requirement for clear and convincing evidence. If the examination process does not include an obviousness analysis, then indeed the burden will be on the patentee to rebut a showing of obviousness by a defendant.

Remedies may also be limited—by making injunctions unavailable, for example, or by capping damages. Other reforms currently being discussed in Congress such as a loser-pays system would also serve to limit litigation and encourage settlement.

Of course, many innovations in the software space are pioneering, and applicants may anticipate that they will have long-term value. A software patent scheme would therefore be an opt-in system, allowing the applicant to select the system deemed most appropriate for the invention.

Instead of trying to force the square peg of software innovations into the round hole of a patent system designed for inventions of a prior era, we should take the opportunity to fashion a system that better serves the needs of the software industry and the public as a whole.

For more information please contact:

Daniel R. Brownstone, 415.875.2358;
dbrownstone@fenwick.com

Reprinted with permission from the July 17, 2014 issue of *The Recorder*.

© 2014 ALM Media Properties, LLC. Further duplication without permission is prohibited. All rights reserved.

THE VIEWS EXPRESSED IN THIS PUBLICATION ARE SOLELY THOSE OF THE AUTHOR, AND DO NOT NECESSARILY REFLECT THE VIEWS OF FENWICK & WEST LLP OR ITS CLIENTS. THE CONTENT OF THE PUBLICATION (“CONTENT”) SHOULD NOT BE REGARDED AS ADVERTISING, SOLICITATION, LEGAL ADVICE OR ANY OTHER ADVICE ON ANY PARTICULAR MATTER. THE PUBLICATION OF ANY CONTENT IS NOT INTENDED TO CREATE AND DOES NOT CONSTITUTE AN ATTORNEY-CLIENT RELATIONSHIP BETWEEN YOU AND FENWICK & WEST LLP. YOU SHOULD NOT ACT OR REFRAIN FROM ACTING ON THE BASIS OF ANY CONTENT INCLUDED IN THE PUBLICATION WITHOUT SEEKING THE APPROPRIATE LEGAL OR PROFESSIONAL ADVICE ON THE PARTICULAR FACTS AND CIRCUMSTANCES AT ISSUE.