



Legally Speaking Software Patents Are Falling Down

Implications of the U.S. Supreme Court's new test for patentable subject matter.

IN *ALICE V. CLS BANK*, the U.S. Supreme Court ruled that a computer-implemented method and system for facilitating settlements of financial transactions was unpatentable subject matter as an abstract idea. In the year or so since that June 2014 decision, many computer-implemented and other software-related patents have been struck down. Moreover, patent infringement lawsuits dropped by 13% in 2014, which a PricewaterhouseCoopers 2015 Patent Litigation Study reports lowered the value of software patent assertions that may be a result of the *Alice* decision. Litigating software patents is now a much riskier proposition for plaintiffs than in the past.

This column discusses the background to the *Alice* decision and the Court's new test for patentable subject matter. It then gives examples of software-related patents that have been invalidated on subject matter grounds when tested in litigation. Some software patents may survive subject matter challenges, but still be at risk of invalidation as indefinite after the Federal Circuit's ruling in *Williamson v. Citrix*.

Background to *Alice v. CLS Bank*

Novel and nonobvious machines, manufactures, compositions of matter, and processes are eligible for patenting so long as these subject matters are properly claimed and vetted through the patent examination process. Because software is a virtual machine and a technological process, some contend that software in-



novations should always qualify as patentable inventions. However, since the Supreme Court's decision in *Gottschalk v. Benson* in 1972, this contention has been, in a word, contentious.

At issue in *Benson* was whether a method for transforming binary coded decimals to pure binary form was eligible subject matter for patenting. One claim covered carrying out this method with the aid of a general-purpose computer; a second claim was for the method more generally, so that a person who carried it out with the aid of paper

and pencil would infringe it. The U.S. Supreme Court ruled that both claims, despite being set forth as a set of steps in method form, failed to recite patentable subject matter.

The Court's rationale for its *Benson* decision is not a model of clarity. The Court was clearly troubled by the abstractness of Benson's method claims, which could be infringed, assuming a patent issued, in a very wide array of applications. The mathematical character of the Benson algorithm raised the specter of a patent impeding use of a

fundamental building block of knowledge. The Court also worried about issuing patents to non-technological innovations that would, in effect, cover mental processes.

The Court revisited the patentability of software-related inventions in 1981 in its *Diamond v. Diehr* decision. *Diehr* upheld the patentability of a rubber-curing process that included a software component. From the mid-1980s and to the late 2000s, the Court of Appeals for the Federal Circuit adopted an expansive interpretation of patentable subject matter. For a time, it appeared everything under the sun made by humans was patentable subject matter as long as it produced a “useful, concrete, and tangible result” (a test that never resulted in the invalidation of any issued patents). Under this standard, hundreds of thousands of software-related and computer-implemented patents were issued in those decades.

Benson Revived in Alice

To make a long story short, the Supreme Court in *Alice v. CLS Bank* de-

cided the *Benson* decision was good law and the Federal Circuit’s expansive interpretation of patentable subject matter was erroneous. (My November 2013 *Communications Legally Speaking* column correctly predicted the Supreme Court would invalidate *Alice*’s patent.)

In *Alice*, the Court decided that abstract methods, such as one facilitating settlements of financial transactions (so that money would only flow when the parties to the transaction had completed their obligations), did not become patentable merely by being carried out through use of a general-purpose computer. Nor did such an abstraction become more patentable if claimed as a machine (or system) to carry out the functions instead of as a process (or method).

To determine if a claimed innovation is patentable subject matter, the Court in *Alice* directed courts to look, first, at the claim language to discern if it recites a law of nature, a natural phenomenon, or an abstract concept, and if it does, then to consider whether the claim as a whole contains additional

elements that “transform the nature of the claim” so that it satisfies patent subject matter requirements.

The Court ruled that *Alice*’s patent recited an abstract concept and the claims as a whole did not satisfy the “something more” second step of the test because they were simply for computer implementations of the abstract concept. However, the Court also implied that software inventions will be patentable subject matter if there is an “inventive concept” in either an improvement in “the functioning of the computer itself” or in “any other technology or technical field.”

Software Patents After Alice

Since *Alice*, the U.S. Patent and Trademark Office (USPTO) has issued revised guidelines on patentable subject matter.^a These guidelines offer numerous examples of software-related inventions, distinguishing among claims that are subject matter eligible under *Alice* and those that are likely not.

The software patents that courts have struck down in the last year or so have generally been those the USPTO issued during the decades in which the Federal Circuit had, in effect, opined that all software-related inventions were patentable subject matter. Had *Alice* been decided in 1990 instead of 2014, many of the computer-implemented and software-related patent claims filed with the USPTO in those years might well have been rejected on subject matter grounds.

The context in which these patents have been falling has largely been litigation in which a challenged firm defends against a claim of patent infringement by attacking the validity of the asserted patent or in which a firm threatened with patent infringement asks a court to declare the asserted patent is invalid on subject matter grounds. Some computer-implemented and software-related patents have survived subject matter challenges, but about three-quarters of the challenges have been successful so far.

Among the many software patents that have fallen when challenged are these:

- Hulu successfully attacked an UL-

^a See <http://1.usa.gov/1gFDI2X>.



**MASTER OF SCIENCE IN
COMPUTER SCIENCE**

With a focus on network security, the MSCS positions students with advanced technical skills essential to today’s programmers.

- One-year or part-time formats
- Evening classes
- Internship and networking opportunities
- Competitive tuition
- Scholarships and financial aid
- Optional test scores

www.merrimack.edu/gradcs
graduate@merrimack.edu • 978-837-3563

**Merrimack
COLLEGE**
NORTH ANDOVER, MA

tramercial patent on a method and system for using advertising as a currency on the Internet in exchange for consumers' getting access to content.

► Costco defeated a patent directed to a computer-implemented system and method for online mortgage shopping that allowed users to assess their borrowing capabilities without revealing their identities that Mortgage Grader, Inc., had obtained from the USPTO.

► Capital One successfully challenged a patent owned by Intellectual Ventures for an online banking method designed to help users create and stick to a budget. The claims were for a method of storing a user profile of specific budgets for each category of transactions the user anticipated, and then transmitting a summary of transactions within each category, so the user could discern if he was on or off budget.

But courts have upheld some software-related patent claims. Intellectual Ventures, for instance, successfully defended a patent aimed at screening computer data for viruses within a telephone network before communicating the data to an end user. Not only was the computerized screening not something a human could do, but the specification described the need for at least three computers to be configured in a specific manner as illustrated by three flowcharts. To practice the claim, it was necessary to imitate the receiving computer's configuration to determine whether a virus in an executable file would infect the receiving computer.

Importance of *Williamson v. Citrix*

Software patents are also being challenged on the grounds the claims are too indefinite to satisfy patent standards. *Williamson v. Citrix* is a case in point. The plaintiff got a patent on a method and system for distributed learning in a virtual classroom environment in which one or more presenters connect with a geographically distributed audience.

(*Williamson* is a trustee of the now bankrupt At Home Corp. that had obtained this patent before the firm went belly up. Adobe, Cisco, IBM Corp., and Microsoft are among the other defendants in *Williamson's*

Complaints about "bad" software patents have been legion among computing professionals.

patent case against Citrix. The patent was an asset of the defunct corporation *Williamson* has been trying to monetize so that holders of bonds in the company could get some compensation for the unfortunate investments they made.)

A trial judge ruled in favor of Citrix and its fellow defendants on all of *Williamson's* patent claims. *Williamson* appealed this loss. Because the trial judge gave an unduly narrow interpretation to some claims, the Federal Circuit overturned the trial judge's ruling on those claims and sent the case back for further proceedings.

However, the Federal Circuit upheld the trial judge's invalidation of eight of *Williamson's* claims because they were too indefinite to satisfy patent standards. The court construed these eight claims as means-plus-function claims, even though the patent did not specifically use the usual means-plus-function language.

Here is some context: U.S. patent law allows patent applicants to claim a means for carrying out a specific function as long as the application discloses particular structures, materials, or acts in the patent specification (which explains the invention in relation to the prior art). This disclosure is necessary to support a means-plus-function patent because the structure, materials, or acts are what makes the claim definite enough so that someone who wanted to do the same function would be able to tell if he or she was infringing the patent or had adopted a non-infringing way to do the same thing.

Under prior Federal Circuit decisions, *Williamson* would have enjoyed a strong presumption the claims at issue were definite enough to survive a challenge because the patent did not use the conventional means-plus-function language. But the Federal Circuit in *Williamson* decided the claims at issue could nonetheless be construed as means-plus-function ones. And because the eight challenged claims lacked the necessary recitation of structures, materials, or acts, the claims were too indefinite to be valid. The Federal Circuit recognized it should not elevate form over substance. Courts can now interpret claims as being means-plus-function ones even if the magic means-plus-function words are missing.

This decision is important because patent lawyers have often sought to broaden the scope of software patents by claiming methods and systems for performing particular functions. By avoiding use of the usual means-plus-function language, the lawyers hoped to attain a very broad scope for the patent, while insulating the patentee from an indefiniteness challenge by omitting the conventional means-plus-function words from the patent claim.

Conclusion

Complaints about "bad" software patents have been legion among computing professionals. Courts seem to be responding to these concerns by developing some new tools to strike down these patents. The most significant tool thus far is the new *Alice* test for patentable subject matter. But the *Williamson* decision may prove to be another important tool with which courts can strike down software patents that are too broad and indefinite. Some may also be struck down for as obvious. Congress is also considering a new package of reforms to curb abuses of the patent system. No silver bullet will solve all software patent problems, but these measures are important steps in the right direction. ■

Pamela Samuelson (pam@law.berkeley.edu) is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California, Berkeley.

Copyright held by author.

Copyright of Communications of the ACM is the property of Association for Computing Machinery and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.