

INSIGHT: Three Indicators a Software Patent May Be Unenforceable

By Joseph Saltiel

Posted May 14, 2019, 4:00 AM

Many conflicting court opinions exist concerning software patents. Rather than trying to reconcile them, Joseph Saltiel, special counsel at Marshall, Gerstein & Borun LLP, clarifies three indicators to look for to determine if a software patent may be unenforceable.

The U.S. patent system has always struggled with how to deal with software. Sometimes software patents were welcomed; sometimes they were not.

In 2014, the U.S. Supreme Court issued its decision in *Alice Corp. Pty. Ltd. v. CLS Bank Int'l* applying a two-part test to determine if a patent was unenforceable for claiming ineligible subject matter. Under this two-part test, a court must first consider whether the claims are directed to a patent ineligible concept such as an abstract idea, and if so, the court should consider whether the claim's other elements transform the nature of the claim into a patent eligible concept.

After *Alice*, software patents were routinely challenged using this test and found to be unenforceable. In the last year, however, courts appear to be more reluctant to quickly find software patents unenforceable. For example, the Federal Circuit's 2018 decision in *Berkheimer v. HP Inc.* softened the effect of *Alice* by holding that the second part of the *Alice* analysis is a question of fact making it more difficult for a defendant to challenge a patent under *Alice* early in a litigation.

Then, earlier in 2019, the PTO issued new guidance on subject matter eligibility that appear to favor some types of software patents.

Despite some recent helpful decisions, some courts will still quickly determine that software patents are unenforceable. For example, in *Walkme Ltd., v. Pendo.io Inc.* (S.D.N.Y. 2019), the court granted an early motion to dismiss of a patent infringement action because the asserted software patent failed the *Alice* test.

Also, in April, the Federal Circuit decided not to follow the PTO guidelines on patent eligibility in a case noting that while it "greatly respects the PTO's expertise on all matters relating to patentability, [courts] are not bound by its guidance." *Cleveland Clinic Foundation v. True Healthcare Diagnostics LLC*.

Thus, five years after *Alice*, there are many conflicting court opinions concerning software patents. Rather than trying to reconcile the conflicting opinions, here are three indicators to look for to determine if a software patent may be unenforceable.

Be Wary of Software Apparatus Claims

A software program is a series of instructions, *i.e.*, a process. Software process claims, however, are executed by customers. Because it is not always practical to enforce a patent against customers, patentees try to reframe their software invention as an apparatus so they can assert their software patent against the companies making the software.

But software is not an apparatus, nor is it any of the other categories of patentable subject matters identified in 35 U.S.C. § 101. In 2014, the Federal Circuit, confirmed that data, *i.e.*, a device profile, not tied to a tangible embodiment of the data, *e.g.*, memory storing the data, does not encompass eligible subject matter either. *Digitech Image Technologies LLC v. Electronics For Imaging Inc.*

And a year later, the Federal Circuit made a similar holding with respect to software, finding software instruction were not patent eligible unless they were recited as a process or tied to a tangible medium. *Allvoice Devs. US LLC v. Microsoft Corp.*

If a claim recites data or software, but it is not recited as a process or have machine elements tied to the software or manipulating the data, that claim is likely not patentable.

Software Claims Should Be Specific

Software patents get into eligibility trouble when they broaden their scope.

For example, a federal court in San Jose, Calif., recently found a patent relating to uploading data unpatentable. *Dropbox Inc. v. Synchronoss Technologies Inc.*

While the specification of the patent provided details on various protocols, servers, and data types involved, the patent claims had three generic steps of creating connections and generating IDs for the connection that were void of details. The court found those claims too generic and abstract.

While adding detail on how to achieve the functionality claimed will limit the scope of the patent claim, it is also more likely to make a software claim eligible for patent protection.

Software Should Be the Star of the Patent

Software and hardware are intertwined. But for a software patent to be eligible, the innovation should be within the software. Relying on hardware as the invention is a sign that software claims will have eligibility problems.

For example, in April, the Federal Circuit affirmed the unpatentability of patents that claimed a process, *i.e.*, software for assisting financial traders execute trades. *Trading Technologies Int'l v. IBG LLC.*

The advantage of this process was not the software itself, but from the underlying hardware that was able to process information quickly, display information in a helpful manner, and execute trader's instructions easily. The claims did not introduce any novel software concepts.

If the software claimed is merely taking advantage of the existing functionality of hardware, it likely may not be patentable.

This column does not necessarily reflect the opinion of The Bureau of National Affairs, Inc. or its owners.

Author Information

Joseph Saltiel is special counsel at Marshall, Gerstein & Borun LLP in Chicago. He is an IP litigator who has represented clients in all types of IP and related matters in courts across the country, the USPTO and the ITC. He regularly counsels clients on IP issues such as licensing, opinions of counsel, NDAs, due diligence and related IP matters.

© 2019 The Bureau of National Affairs, Inc. All Rights Reserved