Is Software Still Patentable?

The last four years have posed significant hurdles to software patents; nevertheless they continue to be filed and allowed.

By James J. DeCarlo and George Zalepa | August 20, 2018 | **The Recorder**

Over the past four years, decisions by the Supreme Court and Federal Circuit, as interpreted by the U.S. Patent and Trademark Office, have had a dramatic effect on software-related inventions. These decisions have focused primarily on what comprises patentable subject matter under 35 U.S.C. § 101 and whether a patent's specification adequately supports the claims. While uncertainty still exists, recent court decisions, coupled with sound prosecution strategies, can be used to bolster a practitioner's arguments before the USPTO and courts.

In *Alice v. CLS Bank* (2014), the Supreme Court applied the now familiar "two-part test" first laid out in *Mayo v. Prometheus* (2012). Under this test, a claim is first analyzed to determine if it is "directed to" an abstract idea. If so, the claim is then analyzed to determine whether it recites "significantly more" than the identified abstract idea or just "routine and conventional" elements. This test was (and is) routinely applied to software claims, and the pendulum of patentability for software inventions post-*Alice* swung firmly towards ineligibility. Nearly all decisions by the Federal Circuit in the immediate aftermath of *Alice* found claims ineligible. Similarly, the USPTO's allowance rate in software-related art units plummeted, and many allowed but not yet issued applications were withdrawn by the USPTO.

While many cases are representative of this period, the decision in *Electric Power Group v. Alstom* (Fed. Cir. 2015) epitomizes the immediate post-*Alice* bias toward ineligibility. There, the Federal Circuit found a claim directed toward power grid monitoring ineligible as the claim broadly described high-level computing operations including “collecting,” “analyzing,” and “presenting” data. While the claims in that case arguably failed to provide a detailed recitation of these steps, this precedent has been used extensively to find software inventions ineligible under § 101 regardless of the level of detail recited.

Fortunately for applicants, the patentability pendulum has begun to rebound. Important decisions such as *McRO v. Bandai* (Fed. Cir. 2016), BASCOM v. AT&T (Fed. Cir. 2016) and *Enfish v. Microsoft* (Fed. Cir. 2016) have provided applicants with new ammunition. Most recent statistics indicate that 2016 was the high-water mark for *Alice* rejections, with the tide turning back in applicants’ favor since.

Enfish, in particular, has provided the clearest guidance. In that case, the court deemed claims directed toward an improved data structure patent-eligible as they improved the functionality of a computing device. The Federal Circuit most recently reaffirmed this problem-solution standard in *Berkheimer v. H.P.* (Fed. Cir. 2018) which involved claims directed toward digitally processing and archiving files. There the court found all claims directed toward an abstract idea (under reasoning similar to Electric Power), but under the second step of *Alice* found certain claims patent-eligible as the specification explicitly stated that claims “improve[d] system operating efficiency and reduce[d] storage costs.” In contrast, the claims deemed ineligible merely recited “routine” components with no clear indication of any improvement in the functioning of a computing device.

The *Berkheimer* case also held that whether an element is routine and conventional is a question of fact. This holding spawned a memo to USPTO examiners advising that any allegation that claim elements are “routine and conventional” now require a factual showing beyond just being known in the art. While this new examination requirement has not yet yielded many decisions, it is being closely watched and should be used as appropriate in rebutting an examiner’s allegations in “step 2” of the Alice analysis.

In parallel with § 101, changes in the interpretation of 35 U.S.C. § 112 have also impacted software-related inventions. Specifically, in *Williamson v. Citrix* (Fed. Cir. 2015), the Federal Circuit applied § 112(f) to a patent claiming a “distributed learning control module.” In construing this term, the court “require[d] that the specification disclose an algorithm for performing the claimed function” performed by the “distributed learning control module.” In finding the term invalid, the court stated that the specification described the functions of the module but failed to disclose “an algorithm for performing the claimed functions.” Thus, the holding in *Williamson* effectively raised the bar for how detailed a specification must be if § 112(f) is invoked.

So while software patents are still viable, their scope has been reined in by *Alice* and *Williamson*. While addressing two distinct statutes, these cases emphasize that a software patent can no longer be drafted at a high-level of functionality. A patent’s specification should clearly articulate both a technical problem addressed by the invention as well as a clear nexus between what is claimed and the technical problem being solved. Given the USPTO’s and courts’ focus on technological improvements, it is vital to go beyond a mere business solution. Applicants should now also presume they will be subject to heightened scrutiny under §§ 101 and 112 and draft a robust specification including as much implementation detail as possible, including detailed flowcharts or even pseudo-code when applicable. The courts and examiners want see a clear description in the specification and claims of how things are done, not merely what is being done.

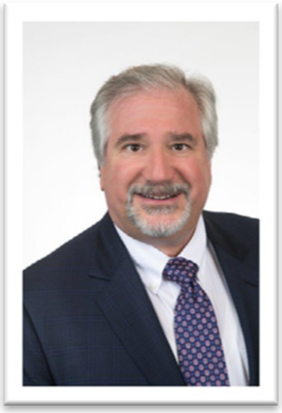
In responding to examiner’s rejections, it is also important for applicants to draw analogies to the slowly growing body of case law finding claims patent eligible. Under the rubric of *Enfish*, the most potent argument for patentability is the ability to identify a clear technical improvement that is recited in the claims and, importantly, is explicitly identified in the specification.

For issued patents, or applications filed pre-*Alice*, such a problem and solution might be difficult to identify or may be missing entirely. In these instances, applicants and patentees may be able to salvage patentability by emphasizing that the specific ordered combination of elements represents significantly more than an allegedly abstract idea, as per the decision in Bascom, or by arguing a lack of factual evidence that the claims elements are "routine and conventional" under *Berkheimer*.

The past four years have posed significant hurdles to software patents; nevertheless, they continue to be filed and allowed. Moving forward, applicants (and inventors) should pay significant attention to the level of detail not just in the claims but also within the written description of the invention. This will pave the clearest path towards not only obtaining patent protection in the software arts but also upholding the validity of such patents in the courts.

About the Authors:

James J. DeCarlo *is a shareholder in Greenberg Traurig's Intellectual Property and Technology Practice. A registered patent attorney and electrical engineer, he is actively involved in virtually all aspects of intellectual property counseling.* **George Zalepa** *focuses his practice on intellectual property and technology matters.*



James J. DeCarlo
decarloj@gtlaw.com

George Zalepa
zalepag@gtlaw.com