

# THE DAILY RECORD

WESTERN NEW YORK'S SOURCE FOR LAW, REAL ESTATE, FINANCE AND GENERAL INTELLIGENCE SINCE 1908

## IP FRONTIERS

### Software patent eligibility – recent development

Software solutions are increasingly implemented in place of human activities, but patenting this type of software can be challenging, especially in view of the Supreme Court's decision in *Alice Corp. Pty. Ltd. v. CLS Bank Int'l.* 134 S. Ct. 2347, 110U.S.P.Q.2d 1976 (2014).

Although that particular decision left little hope for patenting software solutions implemented in place of human activity, because the subject matter can be quickly dismissed in a cursory analysis as an abstract mental process, newer applications of this case by the Federal Circuit and updates of the patent examination guidelines in view of these decisions offer hope in this area. These newer decisions reflect a better grasp of the realities of software and its rapid evolution in and heavy utilization of software/digital technology in just about every area of human activity

Based on the *Alice* decision, when evaluating the patentability of a software claim, the court or USPTO representative makes two inquiries. First, "Is the claim directed to a law of nature, a natural phenomenon, or an abstract idea (*i.e.*, a judicial exception)?" and second, "Does the claim recite additional elements that amount to significantly more than the judicial exception?"

When applying the *Alice* test, the USPTO found many formerly patent-eligible claims fit one of the judicial exceptions, *i.e.*, "an abstract idea," but not "significantly more than the judicial exception," and as a result, the USPTO rejected the claims for being patent-ineligible. During prosecution, far more often than before *Alice*, the USPTO would reject a claim as "an idea 'of itself'" being "an idea standing alone such as an uninstanced concept, plan, or scheme, as well as a mental process (thinking) that "can



By **LUCY SONG**  
Daily Record  
Columnist

be performed in the human mind, or by a human using pen and paper" in the aforementioned chart.

In many cases, although the process could arguably be performed by a human, in reality, it would be ineffective and inefficient so hence, without the use of software, the process would be impossible or inaccurate.

Although the USPTO published Interim Examination Guidance for implementing this test and a chart to assist in identifying abstract ideas, these publications offer an arguably bleak (and still very confusing) picture of the eligibility of software for patenting. Among 27 eligibility examples published by the PTO, 13 examples are marked as abstract ideas, and only 3 examples of computer technologies are non-abstract ideas.

The non-abstract examples include: removing malicious code from email messages, a composite web page, and a system software-BIOS. As software solutions are increasingly implemented in every area of human activity, this group of examples and particularly, the small group of patent eligible examples, cannot accurately represent the full range of software implementations and given the rapid pace of this technological area, attempting to represent the patentability of software with a static group of examples is problematic.

In order to anticipate *Alice* challenges to applications, many practitioners have developed and implemented numerous preemptive anti-*Alice* measures as well as rebuttal strategies to overcome it during the prosecution. Arguably, *Alice* has bur-

dened drafting and prosecuting patent applications for software inventions with lots of patent eligibility technicalities, and shifted significant amount of effort by patent practitioners, from establishing patentability, to establishing patent eligibility of the claims in software patent inventions in view of the USPTO's *Alice* guidance.

Two recent cases decided by the Federal Circuit have, in particular, provided more clarity regarding the patentability of software and arguably, reflect, in the decisions, an understanding of the realities of software technology. These cases, and, optimistically, those that follow, will hopefully focus the patentability analysis of software claims on the substantive patentability of the claims. By applying the decisions in these cases, software that, at least in part, automate a human activity, can be advanced as patentable.

The Federal Circuit's decision in *Enfish, LLC v. Microsoft Corp.* introduced a reality check regarding the subject matter of the claim and the judicial exception (*e.g.*, abstract idea) analysis. In *Enfish*, the Federal Circuit added insight to the initial analysis of whether a claim is directed to an abstract idea. The Court explained that the inquiry is not whether the subject matter involves the abstract idea, but whether it is actually "directed to" the abstract idea. The Federal Circuit clarified that, the "directed to" inquiry applies a stage-one filter to claims, considered in light of the specification, based on whether "their character as a whole is directed to excluded subject matter."

This is an important distinction because many claims will contain a portion that is arguably an abstract idea, especially in a software, but that does not mean that the

*Continued on next page*

*Continued from previous page*

subject matter as a whole is directed to the abstract idea. The Court also stated that in this analysis it relevant to whether the claims are directed to an improvement to computer functionality versus being directed to an abstract idea, and that this inquiry is relevant even at the first step of the *Alice* analysis.

The Federal Circuit's decision in *BASCOM Global Internet Service, Inc., v. AT&T Mobility LLC, et al*, reflects a true understanding of the originality (or lack thereof) of software. The Court sharpened the analysis of finding a claim allowable because it presents "substantially more" under the *Alice* analysis and referred to this "substantially more" as an inventive concept. In the software arena, many inventions are

collections of known elements, which the Court fully appreciated. The Court found that an "arrangement of known, conventional pieces" could include the inventive concept and may therefore be eligible for a patent, and should not be discarded only because the elements in the claim are "known, conventional pieces."

From my experience as a software engineer, I applaud and welcome the rationale for reflecting the reality of software inventions. Unless you are inventing a new programming paradigm, all software programs are written in a "known" programming language, and if we look at the software programs word by word, all software programs may be regarded as "an arrangement of known, conventional pieces" namely keywords and symbols of the programming language.

*Enfish* and *BASCOM* are tools for advocating for the patentability of a claim that at least partially automates an arguably impossible, but according the USPTO, conceivable, human activity. One can argue both the subject matter is not "directed to" the purported judicial exception and/or that this automation is an inventive concept, regardless of whether the components that comprise the invention were known. But more importantly, it is my hope that these cases represent a trend toward evaluating the patentability of software with an understanding of the realities of the technology.

*Lucy is a patent attorney in the law firm of Heslin Rothenberg Farley & Mesiti, P.C. She can be reached in Albany at 518-452-5600 or at hls@hrfmlaw.com.*