

Protecting the value of software patents in the US

Applicants of software patents must constantly navigate an ever-changing legal minefield of potential pitfalls ranging from patentability to enforcement considerations to ensure that the full value of a granted patent can be realised. Brent R. Bellows, Ph.D., Member of Knowles Intellectual Property Strategies, LLC, examines the strategies that exist to protect the value of software patents.

A patent allows its owner to exclude others from making, using or selling the patented invention. Because of this, the filing of a patent application may implicate a wide range of offensive and defensive business goals and represents a powerful tool in any commercial setting. A patent owner that does not strategically position his patent claims to ensure validity and enforcement against potential infringers, however, will not only fail to realize the patent's full value, but also greatly undercut any business goals for filing the patent in the first place.

Because of ever-changing legal, technological and economic landscapes and requirements, many software-related patents thought to be well-positioned and valuable at the time of their grant may no longer be so.

The following discussion provides a non-exhaustive overview of several important issues that any software patent applicant or owner should be aware of in order to best protect the full value of his patents.

Avoid non-patentable subject matter pitfalls

Software-related patents face potential hurdles in meeting the patentable subject matter requirement of 35 U.S.C. § 101¹

due largely to their algorithmic or abstract nature. In the United States, claims directed to 'laws of nature, physical phenomena, and abstract ideas' are not patentable², although 'an application of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection'³. So, while a claim to software *per se* is not patentable⁴, a claim to the application of such software may well be so⁵.

In the *Bilski* case, the Supreme Court, while refusing to expressly rule on the patentability of software-related patents, did indicate that process claims narrowly tailored to the application of abstract ideas, while avoiding pre-empting the idea itself, may be patentable⁶. And even though the Supreme Court eschewed the 'machine-or-transformation' test as the sole test to determine § 101 compliance in *Bilski*, it did reiterate its usefulness as a tool in assisting the analysis⁷. Importantly, several lower courts have subsequently addressed § 101 patentability issues relating directly to software patents. The common theme in these cases appears to be that a software-related claim that ties the underlying application or algorithm to a concrete and limited practical application that avoids broad preemption of a concept is less likely to be an abstract idea and more likely to meet the patentability requirements of § 101 than a claim that is not so tied⁸.

In light of these rulings, applicants must be mindful of how to properly claim software-related inventions to ensure that their claims meet patentability requirements. Applicants should:

- Draft claims that explicitly include the hardware that will interact or be controlled by the software. A claim that involves or is executed by a particular machine or apparatus is less likely to be

drawn to an abstract idea. Applicants should avoid limitations that are merely 'tokens' or that represent 'insignificant, post-solution components'⁹.

- Draft claims that meaningfully manipulate data representing physical or tangible objects. The claims should explicitly identify how the object is transformed from one form into another.
- Draft claims that emphasize the practical or limited nature of the application, avoiding preemption issues.
- Draft additional apparatus and system claims to avoid potential § 101 issues inherent in process claims.
- Draft specifications that clearly identify the functional application of the technology, and how improvements using the technology can be perceived in the real world.

Be aware of potentially differing burdens for proving invalidity

Under United States law, all patents are presumed valid¹⁰. Currently, an alleged infringer challenging the validity of a patent based on invalidating prior art is required to show by 'clear and convincing evidence' that the prior art renders the patent invalid¹¹. This standard applies regardless of whether the prior art was considered by the US Patent and Trademark Office (USPTO) during prosecution of the asserted patent.

Recently, the Supreme Court granted a petition for writ of certiorari in *Microsoft v i4i* to address the proper evidentiary standard to be applied during a validity challenge¹². Of consideration is a 'hybrid' evidentiary standard, wherein 'clear and convincing evidence' would be required to invalidate a patent when the asserted invalidating prior art was considered by the

USPTO during prosecution, while a ‘preponderance of the evidence’ would be required to invalidate the patent when the asserted prior art was not considered.

To the extent that the court adopts this ‘hybrid’ approach, applicants should contemplate strategies for expanding their identification of prior art for citation to the USPTO. This may include performing structured or formalized prior art searches, as well as identifying and citing all known cumulative art. By expanding the scope of prior art cited, applicants would decrease the realm of art that an alleged infringer could assert against the patent, which would inherently increase the value of the patent upon adoption of the ‘hybrid’ standard by ensuring a greater chance of maintaining the heightened standard of validity upon challenge.

Be mindful of the joint infringement trap for process or method claims

To prove direct infringement, a patent owner must show that a single entity practices every element of a valid claim¹³. Joint infringement occurs when no one party practices the asserted claim, but instead the steps of the claim are practiced by different parties¹⁴. Where a joint infringement situation occurs, the claim is directly infringed only if one party exercises ‘control or direction’ over the entire process such that every step is attributable to the controlling party¹⁵. ‘Control or direction’ only exists when there is an agency relationship between the parties who perform the method steps or when one party is contractually obligated to the other to perform the steps¹⁶.

Software-related method claims routinely present joint infringement challenges due to

A software-related claim that ties the underlying application or algorithm to a concrete and limited practical application... is less likely to be an abstract idea and more likely to meet the patentability requirements of § 101 than a claim that is not so tied

their susceptibility to be practice by distributive or decentralized entities. Given the requirement to prove an agency relationship or contractual obligation amongst the practicing parties in these situations, applicants should strive to draft claims from the perspective of a single centralized entity, party, or computer component that cannot be disaggregated (e.g., drafting all steps from the perspective of a single server). Applicants who fail to adequately address joint infringement during claim drafting may be left with valid but unenforceable - and thus valueless - claims¹⁷.

Conclusion

Software patent applicants and owners should manage their portfolios with an eye toward software-specific issues that could detrimentally affect their software patents. Proactive steps may be taken by software patent applicants and owners to avoid these issues. Otherwise, software patent applicants and owners run the risk of unintentionally undermining the value of these business assets.

Brent R. Bellows, Ph.D. Member Knowles Intellectual Property Strategies, LLC
bbellows@kipsllc.com.

1. 35 U.S.C. § 101 provides: Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.
2. Diamond v Chakrabarty, 447 U. S. 303, 309 (1980).
3. Bilski v Kappos, 130 S. Ct. 3218, 3230 (2010) (citing Diamond v. Diehr, 450 U.S. 175, 187 (1981)).
4. See, e.g., Ex Parte Forman, App. No. 2007-1546, Slip op. at 5 (BPAI 21 December 2007) (rejecting a claim to ‘computer code’ not embodied in any tangible medium).
5. See, e.g., Diehr, 450 U.S. at 192-193 (holding that a procedure for molding rubber that included a computer program is within patentable subject

matter).
6. Bilski, 130 S. Ct. at 3229-3231.
7. Id. at 3227.
8. See, e.g., Research Corp. Techs. v Microsoft Corp., 627 F. 3d 859 (Fed. Cir. 2010) (holding that software related claims met § 101 requirements in part because they present ‘functional and palpable applications’); H&R Block Tax Servs. V Jackson Hewitt Tax Serv., Inc., No. 6:08-cv-37 (E.D. Tex., 2 February 2011) (In Magistrate Report and Recommendation, applying a ‘meaningful limits’ test that focuses on preemption to determine compliance with § 101); CLS Bank Int’l v Alice Corp., No: 07-974, 2010 U.S. Dist. LEXIS 23669, *70 (D.D.C. 9 March 2011) (focusing on the extent to which the application of an abstract idea is specific and/or limited to determine whether an invention is patent eligible).
9. See, e.g., H&R Block, Slip op. at 21 (finding that the computer limitation in the claims to be an insignificant, post-solution component, which does not meaningfully limit the claims); Bancorp Servs. v Sun Life Assurance Co., No. 4:99-cv-1073 (E.D. Mo. 14 February 2011) (finding the claims not patentable because they do not ‘narrow the computer implemented method to something more specific than a general purpose computer or recite any specific operations performed that would structurally define the computer’).
10. 35 U.S.C. § 282.
11. See, e.g., American Hoist & Derrick Co. v Sowa & Sons, Inc., 725 F.2d 1350, 1360 (Fed. Cir. 1984).
12. Microsoft Corp. v i4i Limited Partnership, No: 10-290 (U.S. 29 November 2010).
13. BMC Resources, Inc. v Paymentech, 498 F.3d 1373, 1378-79 (Fed. Cir. 2007).
14. For example, if a method claim requires steps A, B, and C, joint infringement occurs if steps A and B are performed by one party, and step C is performed by a second party.
15. See Muniauction, Inc. v Thomson Corp., 532 F.3d 1318, 1329 (Fed. Cir. 2008).
16. Akamai Tech. v Limelight Networks, Inc., 629 F.3d 1311, 1320 (Fed. Cir. 2010).
17. Importantly, alternative theories of contributory or induced patent infringement require that direct infringement has been proven. Aro Mfg. Co. v Convertible Top Replacement Co., 365 U.S. 336, 341-2 (1961) (‘There can be no contributory infringement in the absence of a direct infringement.’).