

REFORMING SOFTWARE CLAIMING

SHUBHA GHOSH*

First of all, it is a real privilege to publish Professor Mark Lemley's article "Software Patents and the Return of Functional Claiming"¹ in the *Wisconsin Law Review*. Not only does Professor Lemley fall squarely in the line of distinguished intellectual property scholars who have given the Kastenmeier Lecture at the University of Wisconsin Law School, but Professor Lemley's article is part of a series of important articles published by the *Wisconsin Law Review*, most recently the symposium on Intergenerational Equity and Intellectual Property in May of 2011.² The many intellectual property articles published in the *Wisconsin Law Review*, of which Professor Lemley's is the most recent example, carry on the spirit of Wisconsin's commitment to understanding society, technology, and legal practice with the goal of social change and reform.

The subject of Professor Lemley's article, software patents, is timely and of social relevance. Software is everywhere, serving as tools to control and direct the flow of information and as modern-day gears and pulleys to operate everyday consumer products. As a critical input to many aspects of our contemporary life, the ability of a company to exclude others from software raises questions about the competitiveness and innovativeness of many industries. Since the 1960s, software patents have been a source of suspicion among those who want to keep software out of the clutches of big business.³ As a result, software patents were subject to intense scrutiny in the 1970s, resulting from the workings of industry lobbying to the creation of copyright for software, which many—including myself—consider an unfortunate result given the long duration of copyright and its ill fit with the functional expression in software.⁴ But as patents recaptured software in the 1980s and 1990s, the problems in a patent regime governing software became apparent as

* Vilas Research Fellow & Professor of Law, University of Wisconsin Law School.

1. Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 WIS. L. REV. 905.

2. Symposium, *Intergenerational Equity and Intellectual Property*, 2011 WIS. L. REV. 103.

3. See JAMES BESSEN & MICHAEL J. MEURER, PATENT FAILURE: HOW JUDGES, BUREAUCRATS, AND LAWYERS PUT INNOVATORS AT RISK 21–24 (2009).

4. See Colleen V. Chien, *Reforming Software Patents*, 50 HOUS. L. REV. 325, 352–59 (2012).

many companies, from start-up to established, found themselves on the defensive side of software patent infringement litigation.⁵

Professor Lemley, as in his many related articles, offers practical and insightful reform proposals to limit the potential and actual harm from software patents.⁶ Instead of categorically removing software from patentable subject matter, he proposes closer judicial scrutiny of software patent claims.⁷ Specifically, he argues courts should limit software patent claims that cover only functions (i.e., what the software does) through strict application of “means-plus-function” claim construction, as codified in Section 112, Paragraph 6 of the Patent Act.⁸ What this means in practice is that software claims that purely claim what a software does should be limited to the structure, material, or acts described in the patent specifications. In other words, courts should limit software patents that claim function through identifying the means by which the function is implemented.

Professor Lemley offers several examples of “functional claiming.”⁹ Let me provide another example. Many gaming patents cover a method of manipulating a device (such as a joystick) in order to simulate swinging a tennis racket or other physical activity.¹⁰ Functional claiming would mean a patent claim for a “method of simulating the swing of a tennis racket.” But this is way too broad even though such a claim could be patentable. To narrow the claim, and its potential effect on all individuals who swing a tennis racket, a means-plus-function claim would prohibit functional claiming by limiting it to particular means, such as the use of a joystick, the use of a particular software algorithm that allows capture of that motion, or perhaps even the use of a tennis racket.

Within the requirements of patentability, Professor Lemley’s proposal is one to reform the enablement requirement, as opposed to the requirements of patentable subject matter, novelty, nonobviousness, or utility.¹¹ The enablement requirement implements the teaching or disclosure requirement of patent: in exchange for the grant of a patent, the inventor must disclose what the invention is in the patent document. With the emphasis on the enablement requirement, Professor Lemley’s

5. *Id.* at 339–43.

6. Lemley, *supra* note 1, at 943–49.

7. *Id.* at 943.

8. *Id.* at 944, 947. See 35 U.S.C. § 112 (2006).

9. Lemley, *supra* note 1, at 919–22.

10. See, e.g., Representing Sub-Events with Physical Exertion Actions, U.S. Patent No. 7,373,387 (filed Apr. 13, 2000). For a general discussion of patents in the area of gaming, see Shubha Ghosh, *Patenting Games: Baker v. Selden Revisited*, 11 VAND. J. ENT. & TECH. L. 871, 885–86 (2009).

11. See Lemley, *supra* note 1, at 946–48.

proposal addresses a frequent criticism of software patents that they are too broad and too inscrutable.¹² As he correctly emphasizes, the proposal is elegant because no change in the law is required.¹³ Courts simply have to apply the strictures of existing law more effectively.

Let me offer two comments about this important proposal. The first is about the substance of the proposal. The second is about its institutional implementation.

Looking to Section 112 as the basis for reforming software patents underscores the central problem of how patent attorneys attempted to navigate the suspicion towards software patenting in the 1970s and 1980s. In response to a series of Supreme Court cases that treated software as a type of mathematical algorithm that could not be patented unless connected to a physical structure like a computer, patent attorneys drafted claims so as to invoke a computer without necessarily limiting the claims to any particular device.¹⁴ Software was treated as a process as distinct from a machine, manufacture, or composition of matter.¹⁵ By comparison, pharmaceutical patents would constitute both composition of matter and process claims, the former applying to the chemical compound and the latter to the production of the chemical compound and to its use in treating a disease.¹⁶ In the 1990s, software as process allowed for the patenting of transformations—that is, the use of software to generate a concrete and tangible result, such as an image on a screen or quantitative information.¹⁷ Functional claiming was the consequence of treating software as process distinct from an unpatentable algorithm but an input into a transformation analogous to a chemical reaction.

Professor Lemley's proposal is to limit the patenting of software as pure function to the particular algorithm or other structure that is the means to attain the function.¹⁸ Filling in the details, failure to disclose such structure in the specifications would presumably invalidate the software claims as indefinite. Providing details of the means would potentially save the claims, but limit them to the specified means and equivalents. The open question is what means will suffice. Professor Lemley rejects the general recitation of a computer or machine as a

12. *See id.*

13. *Id.* at 947–48.

14. *See, e.g., AT&T Corp. v. Excel Commc'ns, Inc.*, 172 F.3d 1352, 1356–58 (Fed. Cir. 1999) (holding patentable a computer program for identifying telephone call recipient's primary carrier).

15. *See Microsoft Corp. v. AT&T Corp.*, 550 U.S. 437, 449 (2007).

16. *See, e.g., Pfizer, Inc. v. Teva Pharm. USA, Inc.*, 429 F.3d 1364, 1369–70 (Fed. Cir. 2005) (example of patent on pharmaceutical compound and process).

17. *See In re Bilski*, 545 F.3d 943, 959–61 (Fed. Cir. 2008).

18. Lemley, *supra* note 1, at 947.

solution.¹⁹ The point of software invention is that software can work across all types of devices. Disclosure of algorithm would require the inventor to relinquish what has traditionally been protected as a trade secret. That would not be a bad result given the goals of disclosure under patent law. But the loss of trade secret protection for software inventions may create pushback to the proposal.

It is instructive to compare the proposal to the approach to software patenting in the European Union, which allows software on patents if there is a technical effect.²⁰ The requirement of showing a technical effect also prevents functional claiming. But the technical effect can be quite general and does not mandate any showing of the structural or other means through which the function occurs.²¹ The United States Patent and Trademark Office (USPTO) applied the technical effects approach in *Bilski v. Kappos*,²² rejecting the application for a method of hedging on the grounds that there was no technical method used in the claim.²³ A technical method rejection is also implicit in the limitation on patenting abstract ideas or mental steps. Arguably, Professor Lemley's approach would be stricter on software patenting by requiring specific disclosure of the algorithm or other structure. But a comparison with the European approach highlights the challenging question of implementation.

A more compelling implementation challenge is the question of institutional choice. Professor Lemley, consistent with his other writings on patent reform, sees the courts as the primary leader in enacting his reforms.²⁴ Perhaps that is the practical solution since legislative patent reform may be too slow and less thoughtful to pursue.²⁵ But the courts, especially the Federal Circuit, may be too divided in applying Section 112 Paragraph 6 with bite. One reason to be skeptical is to go back to the debate over the doctrine of equivalents in the Federal Circuit from a little

19. *Id.*

20. *See, e.g.*, Case T-1173/97, *Computer Program Prod./IBM*, 1999 O.J.E.P.O. 609, 616–17 (the *IBM* case).

21. *See Programs for Computers*, in EUROPEAN PATENT OFFICE, GUIDELINES FOR EXAMINATION IN THE EUROPEAN PATENT OFFICE pt. G, ch. II, § 3.6 at II-7 (2013), available at <http://www.epo.org/law-practice/legal-texts/guidelines.html> (“Such claims . . . may e.g. take the form of a method of operating said apparatus, the apparatus set up to execute the method, or, following T 1173/97 [the *IBM* case], the computer program itself . . .”).

22. 130 S. Ct. 3218 (2010).

23. *Id.* at 3226–27.

24. *See* DAN L. BURK & MARK A. LEMLEY, *THE PATENT CRISIS AND HOW THE COURTS CAN SOLVE IT* 95 (2009).

25. *See generally, e.g.*, Leahy-Smith America Invents Act, Pub. L. No. 112-29 § 1, 125 Stat. 284, 284 (codified in scattered sections of 35 U.S.C. (2011)).

over a decade ago.²⁶ Some judges respect it; others want to eliminate it.²⁷ Because Section 112 incorporates a more restricted version of the doctrine of equivalents, one can predict that the decade-old debate, made dormant by *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*,²⁸ will be re-sparked by the proposal. Of course, the revived debate may be growing pains associated with legal change. More likely, it may serve to squash any positive benefits from the use of Section 112.

The USPTO is another possible source for enacting a stricter Section 112. The *Manual of Patent Examining Procedure* has an extensive discussion of Section 112 with application to software.²⁹ Although this discussion seems to track the Federal Circuit case law, the USPTO also attempts to synthesize and interpret the case law in a more cohesive and effective way.³⁰ The type of reform Professor Lemley seeks, I think, could best be implemented through an administrative agency. The obstacle is whether the USPTO has such rule-making and interpretative capacity. Courts give the USPTO a reduced form of deference.³¹ One recent strand of scholarship is to advocate heightened deference to the USPTO.³² Some scholars even argue that the America Invents Act invites *Chevron* deference.³³ Professor Lemley's important and incisive proposal may ultimately reveal the inadequacy of the courts to deal with patent reform. The need for agency-based legal reform is likely the next phase in the policy debate and undoubtedly fodder for another perceptive and engaging paper from the pen of Professor Lemley. Dare I propose a *Kastenmeier II*?

26. See *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 234 F.3d 558, 564 (Fed. Cir. 2000).

27. *Id.* at 566 (offering Judge Schall's critical view); *id.* at 630 (Newman, J., concurring and dissenting) (offering a defense of the doctrine of equivalents).

28. 535 U.S. 722 (2002).

29. U.S. PATENT & TRADEMARK OFFICE, *MANUAL OF PATENT EXAMINING PROCEDURE* § 2181 (8th ed., rev. 2012).

30. *Id.* at § 2181 II.B.

31. See *Dickinson v. Zurko*, 527 U.S. 150, 154–55 (1999) (adopting a substantial evidence standard for review of agency fact finding).

32. See, e.g., Stuart M. Benjamin & Arti K. Rai, *Who's Afraid of the APA? What the Patent System Can Learn from Administrative Law*, 95 GEO. L.J. 269, 270–74 (2007).

33. See Melissa F. Wasserman, *The Changing Guard of Patent Law: Chevron Deference for the PTO*, 54 WM. & MARY L. REV. 1959, 1966 (2013).