Journal of Business & Technology Law

Volume 10
Issue 1 *The State of Concussions: Protecting Athletes through Advances in Law, Public Health, and Science*

Article 9

Application of Functional Claiming Limitations: The Practical Effects on Software-Related Patents

Ryan Steidl

Ryan Steidl*

Application of Functional Claiming Limitations: The Practical Effects on Software–Related Patents

"*That ideas should freely spread from one to another over the globe, for the moral and mutual instruction of man, and improvement of his condition, seems to have been peculiarly and benevolently designed by nature, when she made them, like fire, expansible over all space, without lessening their density in any point, and like the air in which we breathe, move, and have our physical being, incapable of confinement or exclusive approbation.*"[1]

Software-related patents[2] have become the subject of considerable scrutiny as the rate and volume of patents involved in litigation continues to increase rapidly, especially among patents with claim language that defines elements by their function rather than their structure.[3] In an analysis of 500 representative lawsuits

* J.D., University of Maryland Francis King Carey School of Law, 2016; M.B.A., University of Maryland Robert H. Smith School of Business, 2016; B.S., Pennsylvania State University, 2009. The author would like to thank Professor James Grimmelmann for all of his insights, mentorship, and encouragement in writing this Comment. For the immeasurable support provided by all the author's family, words alone cannot express adequate thanks. The author would also like to thank Morrie Amitay for providing the foundational opportunities that developed such boundless confidence and determination to pursue considerable ambitions. Finally, the author would like to thank his many friends—those who are practically family, members of the *Journal of Business and Technology Law*, and fellow members of the Carey School of Law and Smith School of Business—for their cheer and commiseration, relentless optimism in the face of adversity, and steadying presence in all waters.

1. Letter from Thomas Jefferson to Isaac McPherson (Aug. 13, 1813) (on file with Founders Online, National Archives), *available at* http://founders.archives.gov/documents/Jefferson/03-06-02-0322.

2. The term "software patent" is used most often in a loose and colloquial manner. Whenever claims involve a computer, processing device, or electronic technology, typically this generic term is used to describe the patent. However, nonspecific descriptors create many complications—for instance, in practice, lack of uniform terminology has made it difficult, if not impossible, to perform adequate software-related patent searches. *See, e.g.*, Christina Mulligan & Timothy B. Lee, *Scaling the Patent System*, 68 N.Y.U. Ann. Surv. Am. L. 289, 297–305 (2012), *available at* http://digitalcommons.law.uga.edu/fac_artchop/913.

3. *See* Government Accountability Office, *Intellectual Property: Assessing Factors That Affect Patent Infringement Litigation Could Help Improve Patent Quality*, Aug. 22, 2013, *available at* http://www.gao.gov/products/GAO-13-465. *See also* Dennis Crouch, *Functional Claim Language in Issued Patents*, Patently-O Blog

from 2007 to 2011, the Government Accountability Office found a 129 percent increase in the number of defendants involved in patent infringement litigation, with software-related patent issues accounting for 89 percent of the increase.[4] Despite the many reasons for such proliferation,[5] software-related patents present challenges that are distinct from other fields of industry, including "problems stemming from its mathematical properties, the structure of the software market, and the importance of interoperability."[6] Although many ideas have been proposed for resolving the challenges of how to manage software-related patents,[7] this Comment focuses on the particular approach that has been advocated by Professor Mark Lemley.[8]

Lemley has been an advocate for stricter application of functional claiming limitations under 35 U.S.C. § 112(f)[9] and restricting patent claims that define elements by what they "do" rather than what they "are."[10] While the effectiveness of Lemley's proposal is uncertain until it can be tested before judges who understand the details,[11] it is still possible to gain a sense of what the practical effects may be from adopting his approach. Accordingly, this Comment applies stricter functional claiming limitations to a survey of previously litigated patents, analyzes the claims, and explains what outcomes might occur.

In order to provide full context, this Comment also examines some of the related issues associated with software-related patents. Part I discusses background topics including problematic issues with software-related patents, policies, and proposed resolutions for such problems—this part discusses some of the major difficulties with the current state of software patents,[12] suggested solutions others have set forth,[13] and some of the inadequate guidance that has been provided by the courts

(Jan. 23, 2014), *available at* http://patentlyo.com/patent/2014/01/functional-language-patents.html; ROBERT C. FABER, FABER ON MECHANICS OF PATENT CLAIM DRAFTING 3-111 (6th ed. 2008).

4. *Intellectual Property: Assessing Factors That Affect Patent Infringement Litigation Could Help Improve Patent Quality*, *supra* note 3, at 14.

5. *Id*. at 28–35.

6. BEN KLEMENS, MATH YOU CAN'T USE: PATENTS, COPYRIGHT, AND SOFTWARE 4 (Brookings Inst. Press 2006). Klemens' book is a phenomenal resource on software-related patents—it provides an excellent in-depth examination of the many issues and conflicts between software/mathematics, law, and economics.

7. *See infra* Part I.B.

8. Professor Mark Lemley is the William H. Neukom Professor of Law at Stanford Law School, the Director of the Stanford Program in Law, Science and Technology, Director of Stanford's LLM Program in Law, Science and Technology, and a founding partner of Durie Tangri LLP.

9. Herein referred to alternatively as "Section 112(f)" or simply "112(f)."

10. *See infra* Part II.B; *see also* FABER, *supra* note 3.

11. Mike Masnick, *Fixing Software Patents by Actually Applying Exiting Patent Law*, TECHDIRT.COM (Sept. 20, 2012), www.techdirt.com/articles/20120915/23461020394/fixing-software-patents-actually-applying-existing-patent-law.shtml. *See also* Clyde Findley, *Hope for Software Patents*, CLOUDIGYLAW.COM, (Nov. 13, 2013), http://cloudigylaw.com/hope-software-patents/.

12. *See infra* Part I.A.

13. *See infra* Part I.B.

so far.[14] Part II provides an overview of Lemley's work, including historical use of functional claiming, current treatment of software-related patents, and potential benefits of Lemley's proposed solution.[15] Part III examines and explains how patents would be analyzed under stricter 112(f) functional claiming limitations,[16] and whether the claims would be able to meet those standards.[17] Finally, the conclusion reached here is that although many software-related patents would not survive strict application of 112(f), such an approach would not be too burdensome or overbroad, and instead would hold software-related patents to a much-needed higher standard.[18]

I. PROBLEMS, POLICY, AND PROPOSED SOLUTIONS

A. Problematic Issues with Software-Related Patents

Many practitioners—in law, business, technology, and other fields—have been critical of software-related patents due to a variety of problematic issues. Some of the most basic issues involve the patent system's incompatibility with the scale of software-related industries, the stockpiling of patents by both practicing and non-practicing entities, as well as the unique qualities of software-related patents that distinguish them from other intellectual property.

The sheer size and complexity of software-based industries presents daunting challenges for navigating the current patent system.[19] For perspective, looking at an estimated 600,000 firms creating software, and 40,000 software patents issued each year (and growing), the resulting number of patent-firm pairs that could potentially infringe comes in at around 24 billion.[20] Consequently, there is no cost-effective or time-effective way for businesses or firms to discover and ensure noninfringement of all existing patents.[21] Not surprisingly, it is common practice to simply ignore patents altogether[22] and justify such actions with excuses of: length of time it takes to settle negotiations, unwillingness of patent holders to offer reasonable terms, invalidity of many patents, and the prohibitively high numbers of licenses that may need to be obtained.[23]

The term "patent thicket" aptly describes the many problems associated with scale—it is defined as "a dense web of overlapping intellectual property rights that a

14. *See infra* Part I.C.
15. *See infra* Part II.
16. *See infra* Part III.A.
17. *See infra* Part III.B.
18. *See infra* Part IV.
19. *See* Mulligan & Lee, *supra* note 2, at 293.
20. *Id.* at 304.
21. *Id.* at 307.
22. Mark A. Lemley, *Ignoring Patents,* 2008 MICH. ST. L. REV. 19, 21 (2008).
23. *Id*. at 25–29.

company must hack its way through in order to actually commercialize new technology. With the cumulative effects of innovation and multiple blocking patents, stronger patent rights can have the perverse effect of stifling, not encouraging, innovation."[24] Although there can be different types of patent thickets, many thickets stifle innovation due to the adverse interrelationship and assertions of existing patent rights.[25]

Using patent thickets for leverage has led to multi-billion dollar arms races and various forms of gamesmanship[26] given that exclusive rights and uncontested power without competition in the marketplace[27] provides strong incentives for pursuing a domineering type of strategy. For larger entities, it is not in their interests to completely exclude others from a market.[28] Many large businesses flex their expansive portfolios and positions to "negotiate" licensing deals with smaller entities[29]—particularly innovative businesses that have recently become successful.[30] Many businesses must simply assume that all software will somehow infringe on an existing patent, and so concerns must necessarily shift from absolute avoidance to minimizing the probability that a patent holder will pursue action, or minimizing the licensing fees that will be paid.[31] Yet whether used as bargaining chips, or for

24. Carl Shapiro, *Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting*, 1 INNOVATION POL'Y & ECON. 119, 120 (2000).

25. *See, e.g.*, James Bessen, *Patent Thickets: Strategic Patenting of Complex Technologies* (Bos. Univ. School of Law, Working Paper, 2003), *available at*: http://ssrn.com/abstract=327760.

26. Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 WIS. L. REV. 905, 931–32 (2013). *See also* KLEMENS, *supra* note 6, at 80–90.

27. *See* Andrew Beckerman-Rodau, *Patents Are Property: A Fundamental But Important Concept*, 4 J. BUS. & TECH. L. 87, 88, 90 (2009), *available at* http://digitalcommons.law.umaryland.edu/jbtl/vol4/iss1/4.

28. KLEMENS, *supra* note 6, at 83–87.

29. The types of threats posed by patent thickets is evident in an anecdote from Gary L. Reback when he represented Sun Microsystems in the 1980s. IBM had accused Sun of infringing on seven patents, threatening the then-small company with a massive lawsuit, and demanding a meeting to discuss the claims. According to Reback, even though the IBM legal team accepted Sun's conclusions that no court would likely find infringement, they replied coldly: "Maybe you don't infringe on these patents. But we have 10,000 U.S. patents. Do you want us to go back to our headquarters and find seven you do infringe on, or do you want to make this easy and just pay us $20 million?" After some negotiations, Sun eventually cut IBM a check. *See* Gary L. Reback, *Patently Absurd*, FORBES, (June 24, 2002), *available at* http://www.forbes.com/asap/2002/0624/044.html.

30. Twitter's update to its S-1 in November 2013 provides an example of the ongoing practice of leveraging patent portfolios against businesses that have gained attention from recent successes. Twitter disclosed that IBM had sent a letter alleging infringement of at least three U.S. patents and offered an invitation to "negotiate a business resolution of the allegations." Like Sun Microsystems, it appears Twitter is prepared to defend itself against these kinds of claims as it becomes a bigger target. Of note, Twitter's disclosure raises concerns that are likely shared by other growing businesses: "From time to time we may introduce new products and services, including in areas where we currently do not have an offering, which could increase our exposure to patent and other intellectual property claims from competitors and non-practicing entities." *See* Ingrid Lunden, *IBM Claims Twitter Infringes On At Least 3 Of Its Patents, According To Twitter's Latest S-1 Update*, TECHCRUNCH.COM (Nov. 4, 2013), http://techcrunch.com/2013/11/04/ibm-claims-twitter-infringes-on-at-least-3-of-its-patents-according-to-twitters-latest-s-1-update/.

31. It has been calculated that it is mathematically impossible to avoid infringement: "[I]t would take at least 2,000,000 patent attorneys, working full time, to consider whether all these software-producing firms have infringed any of the software patents issued in a typical year. Even if firms wanted to hire that many attorneys,

licensing purposes, the core function of software-related patents has seemingly become less about innovation and more about "business strategy, gaming, squeezing players out, and preventing people from wanting to take risks."[32]

Patent leveraging is also done by entities that own patents but do not directly use them to produce goods or services, instead asserting the patents against companies that do produce such goods and services.[33] These non-practicing entities ("NPEs") are known by the more pejorative term "patent troll," despite the distinction between the two.[34] The costs of NPE patent assertions are substantial, estimated at potentially $29 billion per year in direct costs,[35] with annual wealth losses from NPE lawsuits estimated at about $80 billion.[36] Congress has taken some interest in patent litigation and fixing gaps in the 2011 America Invents Act—such as the failure to rein in abuses by NPEs—but any changes will most likely be slow to occur.[37]

In addition to the various issues with scale and stockpiling of software-related patents, such patents also possess unique qualities that courts have often inadequately distinguished from other patentable subject matter.[38] The most overlooked consideration is arguably that "a sufficiently detailed description of a computer program is the program itself, so it is sometimes difficult to distinguish between the idea and its implementation."[39] Secondly, the courts have done little to resolve the contradiction that mathematics is unpatentable subject matter,[40] yet "in a literal sense, at its most basic level software is nothing more than a piece of mathematics."[41] Finally, courts have been slow to recognize that "software patents are a restriction not only on competitors but on a wide array of computer users."[42]

they couldn't; there are only 40,000 registered patent attorneys and agents in the United States." *See* Timothy B. Lee & Christina Mulligan, *Opinion: The Problem With Software Patents? They Don't Scale*, ARSTECHNICA (Mar. 8, 2012), *available at* http://arstechnica.com/tech-policy/2012/03/opinion-the-problem-with-software-patents-they-dont-scale/; *see also* Mulligan & Lee, *supra* note 2, at 304–305; *c.f.* KLEMENS, *supra* note 6, at 85 (Licensing fees are major sources of revenues for larger entities. For instance, from 1993 to 2002 IBM had 22,357 patents granted to it and received $10 billion in licensing fees.).

32. *Hearing on Competition and Intellectual Property Law and Policy in the Knowledge-Based Economy*, Fed. Trade Comm'n, 138, (Mar. 20, 2002) (testimony of Edward. J. Black), *available at*: www.ftc.gov/opp/intellect/020320trans.pdf.

33. James Bessen & Michael J. Meurer, *The Direct Costs from NPE Disputes* 3 (Bos. Univ. School of Law, Working Paper No. 12-34, 2012), *available at* http://www.bu.edu/law/faculty/scholarship/workingpapers/2012.html.

34. Not all NPEs are patent trolls. For instance, a university may be an NPE but no one would consider a university to be a "patent troll" in the traditional sense. *See* David A. Boag, *Rep. Bob Goodlatte's 43-Day Assault on the Patent Troll*, TECHCRUNCH.COM (Dec. 15, 2013), http://techcrunch.com/2013/12/13/rep-goodlattes-43-day-assault-on-the-patent-troll/.

35. Bessen & Meurer, *supra* note 33, at 19.

36. *Id.* at 5.

37. Boag, *supra* note 34.

38. KLEMENS, *supra* note 6, at 4–5.

39. *Id*.

40. *See*, *e.g.*, MANUAL OF PATENT EXAMINING PROCEDURE § 2106 (2014).

41. KLEMENS, *supra* note 6, at 4–5.

42. *Id*.

Although these three points are not an exhaustive list of software's unique qualities, they are important considerations that should be accounted for in any proposed or adopted reforms to current laws.

B. Proposed Solutions to Software-Related Patents

There have been many suggestions put forth for how to "solve" the various problems associated with software-related patents. In 2012, a conference entitled "Solutions to the Software Patent Problem" brought together some of the leading ideas and different approaches that might be considered for more widespread adoption.[43] Proposals have ranged from outright abolishment of the software patent category altogether, to greater enforcement of existing legislation, to expanded use of independent inventorship considerations.

The most sweeping proposal by far has sought an end to all software patents.[44] Groups have petitioned the White House to "pursue software patent abolition,"[45] while those involved in areas of academia[46] and entrepreneurship[47] alike have also voiced support for eliminating software-related patents.[48] After New Zealand approved a general ban on software patents,[49] some have called for the United States to follow suit.[50] Yet outright abolishment has been criticized as an overbroad approach and is unlikely to actually occur.[51]

A more basic proposal has suggested heightening the standards of review at the U.S. Patent and Trademark Office in order to weed out bad patents.[52] In a comparison to the bioinformatics field, it has been asserted that more stringent

43. *See* Solutions to the Software Patent Problem, Santa Clara Univ., http://law.scu.edu/hightech/2012-solutions-to-the-software-patent-problem/.

44. *See, e.g.*, Lemley, *supra* note 26, at 936–37 n.132.

45. *See* Dennis Crouch, *We the People: Petitioning President Obama to End Software Patents*, PATENTLY-O BLOG (Nov. 2, 2011), http:// www.patentlyo.com/patent/2011/11/we-the-people-petitioning-president-obama-to-end-software-patents.html.

46. *See generally* Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025 (1990).

47. *See generally* Vivek Wadha, *Why We Need to Abolish Software Patents*, TECHCRUNCH.COM (Aug., 7, 2010), http://techcrunch.com/2010/08/07/why-we-need-to-abolish-software-patents/.

48. *See, e.g.*, Colleen V. Chien, *Reforming Software Patents*, 50 HOUS. L. REV. 325, 352–53 (2012) (noting proposals to abolish software patents as a potential reform approach for reducing patent numbers).

49. Lucy Craymer, *New Zealand Ends Patents for Basic Software*, WALL ST. J., Sept. 1, 2013, *available at* http://blogs.wsj.com/digits/2013/09/01/new-zealand-ends-patents-for-basic-software/.

50. *See, e.g.*, Timothy B. Lee, *New Zealand Just Abolished Software Patents. Here's Why We Should, Too.*, WASH. POST, Aug. 29, 2013, *available at* http://www.washingtonpost.com/blogs/the-switch/wp/2013/08/29/new-zealand-just-abolished-software-patents-heres-why-we-should-too/. *See also* Ben Taylor, *It's Time to Outlaw Software Patents*, AMERICANTHINKER.COM (Sept. 7, 2013), *available at* http://www.americanthinker.com/2013/09/its_time_to_outlaw_software_patents.html.

51. *See, e.g.*, Lemley, *supra* note 26, at 937.

52. *Id.* at 938–39. *See also* Arti Rai, *Let's Tame Software Patent Claims: Lessons From Bioinformatics*, WIRED.COM (Nov. 20, 2012), *available at* http://www.wired.com/opinion/2012/11/software-patents-bioinformatics/.

scrutiny should lead to an increase in rejection on obviousness grounds, thereby increasing the quality of patents granted overall.[53] One of the arguments in support of this approach is that it could be implemented immediately under existing law.[54] Yet some counter-argue pessimistically that reforms made at an administrative level would ultimately be ineffective at improving patent quality.[55]

More nuanced ideas propose greater leniency or allowances with regards to independent inventorship.[56] One suggested approach would be to change the current rules and allow independent invention to be asserted as a defense to infringement.[57] Another approach would be to judge obviousness by whether multiple inventors have identified the same results at approximately the same time.[58] Lastly, another approach would be to comparatively weigh the costs of discovering a patentee's invention against the costs of independently inventing that same invention.[59]

C. Guidance and Precedent from the Federal Circuit and U.S. Supreme Court

The above proposals represent only a small fraction of the various approaches that have been suggested, and have come about in part because of inadequate guidance from the courts. In particular, the Federal Circuit has faced substantial criticism for its unhelpful rulings and lack of clear guidance on resolving problematic aspects of software-related patents,[60] and has even criticized itself for its own discordant results.[61] The Federal Circuit arguably reached its lowest point when it sat en banc

53. Rai, *supra* note 52.

54. *Id*. *See also* Eric Goldman, *How to Fix Software Patents (Part 3 of 3)*, FORBES (Dec. 12, 2012), *available at* http://www.forbes.com/sites/ericgoldman/2012/12/12/how-to-fix-software-patents/.

55. R. Polk Wagner, *Understanding Patent-Quality Mechanisms*, 157 U. PA. L. REV. 2135, 2163–65 (2009).

56. *See infra* notes 57–59 and accompanying text.

57. *See, e.g.*, Samson Vermont, *Independent Invention as a Defense to Patent Infringement*, 105 MICH. L. REV. 475 (2006).

58. *See, e.g.*, John Duffy, *Let's Get Rid of Kludgy Patent Fixes and Define the Non-Obvious*, WIRED.COM (Nov. 16, 2012), *available at* http://www.wired.com/opinion/2012/11/lets-get-rid-of-kludgy-patent-fixes-and-define-the-non-obvious/. *See also* Michael Abramowicz & John F. Duffy, *The Inducement Standard of Patentability*, 120 YALE L.J. 1590, 1614 (2011).

59. *See* Goldman, supra note 54. *See generally* Samson Vermont, *Basing Patent Remedies on Harm to the World Instead of Harm to the Patentee*, (unpublished manuscript) (on file with Stanford Univ.), *available at*: http://www.stanford.edu/dept/law/ipsc/Paper%20PDF/Vermont,%20Samson%20-%20Paper.pdf.

60. *See* James Grimmelmann, *If Our Patent Court Screws Up Slipper Patents, How Can It Rule Sensibly on Smartphones?* WASH. POST, Sept. 24, 2013, *available at* http://www.washingtonpost.com/blogs/the-switch/wp/201.
3/09/24/if-our-top-patent-court-screws-up-slipper-patents-how-can-it-rule-sensibly-on-smartphones/. *See also* James P. Muraff & Kevin Cukierski, *Predictability Still Needed From Divided Federal Circuit on Software Patents*, NATIONAL LAW REVIEW (Sept. 16, 2013), http://www.natlawreview.com/article/predictability-still-needed-divided-federal-circuit-software-patents.

61. *See* CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1314 (Fed. Cir. 2013)("Our court is irreconcilably fractured over these system claims and there are many similar cases pending before our court and the district courts.") (Moore, J., dissenting in part); *see also id.* at 1321 ("The court . . . hoped to ameliorate [many uncertainties] by providing objective standards for section 101 patent-eligibility. Instead we have

to hear the case of *CLS Bank v. Alice Corp.* and issued several opinions devoid of agreement other than to find the claims invalid.[62]

Although the U.S. Supreme Court's decisions have been more informative than those of the Federal Circuit, its opinions have not fully resolved the ongoing issues with software-related patents. For instance, although the Court heard *Bilski v. Kappos* in 2010,[63] enduring confusion led the Court to revisit many of the same issues in *Alice Corp. v. CLS Bank* just three years later.[64] Despite the lack of an apparently definitive opinion on software-related patents, several of the Supreme Court's precedents are important to note for insight into what approaches the Court has taken in the past, and what guiding considerations it may rely on in the future.

The *Benson-Flook-Diehr* trilogy of cases in particular reveals the evolution of the Supreme Court's treatment of software-related patents and where the line should be drawn for eligibility.[65]

In *Gottschalk v. Benson* the Court ruled that an invention which is nothing more than a broad mathematical idea cannot be patentable.[66] Although the patent claims required a computer for implementation, the Court did not find this to be a meaningful limitation, since "[t]he mathematical formula involved here has no substantial practical application except in connection with a digital computer, which . . . would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself."[67]

Six years after *Benson*, the Court again asserted limitations for software patents when it ruled in *Parker v. Flook* that merely adding token "post-solution activity" was insufficient for circumventing the bar on claiming a mathematical formula.[68] As reiterated by the *Bilski* Court: "*Flook* stands for the proposition that the prohibition against patenting abstract ideas 'cannot be circumvented by attempting to limit the use of the formula to a particular technological environment' or adding 'insignificant postsolution activity.'"[69]

propounded at least three incompatible standards, devoid of consensus, serving simply to add to the unreliability and cost of the system of patents as an incentive for innovation. With today's judicial deadlock, the only assurance is that any successful innovation is likely to be challenged in opportunistic litigation, whose result will depend on the random selection of the panel.") (Newman, J., concurring in part, dissenting in part).

62. *Id.*

63. Bilski v. Kappos, 130 S. Ct. 3218 (2010).

64. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014).

65. *See, e.g.*, KLEMENS, *supra* note 6, at 53.

66. Gottschalk v. Benson, 93 S. Ct. 253, 257 (1972).

67. *Id.* at 257.

68. Parker v. Flook, 437 U.S. 584, 585, 596 (1978).

69. Bilski v. Kappos, 130 S. Ct. 3218, 3230 (2010) (*citing* Diamond v. Diehr, 450 U.S. 175 (1981)). *See also* Parker v. Flook, 437 U.S. 584, 599 (1978) ("The notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance. A competent draftsman could attach some form of post-solution activity to almost any mathematical formula; the Pythagorean theorem would not have been patentable, or partially patentable, because a patent

Three years after *Flook*, the Supreme Court heard *Diamond v. Diehr*.[70] The Court conceded that although "[the claimed] process admittedly employs a well-known mathematical equation," in contrast with *Benson* and *Flook*, the claims did "not seek to preempt the use of that equation."[71] The Court noted that while "abstract ideas, laws of nature, or mathematical formula" may not be patented, "an *application* of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection."[72]

More than 30 years later, the Supreme Court relied on the foundations of the *Benson-Flook-Diehr* trilogy when it issued its decision in *Bilski v. Kappos*.[73] The Court found that the patent claims could not receive protection since "[they] attempt to patent the use of the abstract idea of hedging risk in the energy market and then instruct the use of well-known random analysis techniques to help establish some of the inputs into the equation."[74] Ultimately, the Court asserted that "[r]ather than adopting categorical rules that might have wide-ranging and unforeseen impacts, the Court resolves this case narrowly on the basis of this Court's decisions in *Benson, Flook,* and *Diehr*."[75] However, *Bilski* was largely seen as establishing very "little by way of clear and cogent parameters for method-claim eligibility, creating a quagmire for patent examiners and patent practitioners in their roles as primary gatekeepers of the patent system."[76]

In 2013, the Federal Circuit heard the case of *CLS Bank v. Alice Corp.*,[77] which some hoped would establish a more administrable framework for software-related

application contained a final step indicating that the formula, when solved, could be usefully applied to existing surveying techniques.").

70. Diamond v. Diehr, 450 U.S. 175 (1981).

71. *Id*. at 187.

72. *Bilski*, 130 S. Ct. at 3230 (*citing Diehr*, 450 U.S. at 187 (emphasis original)). "We recognize, of course, that when a claim recites a mathematical formula (or scientific principle or phenomenon of nature), an inquiry must be made into whether the claim is seeking patent protection for that formula in the abstract. A mathematical formula as such is not accorded the protection of our patent laws, . . . and this principle cannot be circumvented by attempting to limit the use of the formula to a particular technological environment. . . . Similarly, insignificant post-solution activity will not transform an unpatentable principle into a patentable process. . . . To hold otherwise would allow a competent draftsman to evade the recognized limitations on the type of subject matter eligible for patent protection. On the other hand, when a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect (*e.g.*, transforming or reducing an article to a different state or thing), then the claim satisfies the requirements of § 101." *Diehr*, 450 U.S. at 191–93.

73. *Bilski*, 130 S. Ct. at 3229.

74. *Id*. at 3231.

75. *Id*. at 3229.

76. *See* Abby Bhattacharayya, *Implementation, or the Possible Lack Thereof, of the Bilski Supreme Court Decision*, 6 J. BUS. & TECH. L. 103, 103 (2011), *available at*: http://digitalcommons.law.umaryland.edu/jbtl/vol6/iss1/5. *See also infra* notes 78 and 79, and accompanying text.

77. CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269 (Fed. Cir. 2013) (per curiam).

patents[78] since the *Bilski* decision did not provide much resolution for practitioners.[79] The case concerned patents owned by Alice Corporation involving the use of an escrow system to manage risk, and "relate[d] to a computerized trading platform used for conducting financial transactions in which a third party settles obligations between a first and a second party so as to eliminate 'counterparty' or 'settlement' risk."[80] The en banc court could not reach a consensus,[81] and among the opinions was the alarmist warning that "if all of these claims . . . are not patent-eligible, this case is the death of hundreds of thousands of patents, including all business method, financial system, and software patents as well as many computer implemented and telecommunications patents."[82]

Not surprisingly, the *CLS v. Alice Corp.* decision was appealed to the Supreme Court. In a unanimous decision striking down the patents, the Court reiterated that the *Benson-Flook-Diehr* trilogy demonstrates that "mere recitation of a generic computer cannot transform a patent-ineligible abstract idea into a patent-eligible invention."[83] The Court asserted that it is irrelevant that a computer is a physical object, since "if that were the end of the § 101 inquiry, an applicant could claim any principle of the physical or social sciences by reciting a computer system configured to implement the relevant concept," and as a result the determination of patent eligibility would "depend simply on the draftsman's art."[84] Although practitioners have reacted with more optimism about this decision than others from the Supreme Court—and although this decision has already had more of an immediate effect—there are still reservations that it may only be a matter of time before circumventions are discovered and exploited.[85]

78. *See, e.g.*, Tony Dutra, *CLS v. Alice en banc: Finally resolving post-Bilski mess?*, BLOOMBERG BNA (Oct. 16, 2012), http://www.bna.com/cls-alice-en-b17179870238/.

79. *See, e.g.*, Steve Lohr, *Bilski Ruling: The Patent Wars Untouched*, N.Y. TIMES BITS BLOG (June 28, 2010, 7:31 PM), http:// bits.blogs.nytimes.com/2010/06/28/bilski-ruling-the-patent-wars-untouched/.

80. *CLS Bank Int'l.*, 717 F.3d at 1274.

81. *Id.* at 1282. A plurality of five judges argued that the relevant inquiry should involve identifying preemption concerns, and performing an analysis where "the balance of the claim [is] evaluated to determine whether it contains additional substantive limitations that narrow, confine, or otherwise tie down the claim so that, in practical terms, it does not cover the full abstract idea itself." *Id.* Another plurality of four judges asserted that, "[t]he relevant inquiry must be whether a claim includes *meaningful* limitations restricting it to an application, rather than merely an abstract idea." *Id.* at 1299 (emphasis in original). And in another plurality opinion, concern was expressed that with the current interpretation of § 101, application of the abstract idea exception had been given "staggering breadth to what is meant to be a narrow judicial exception." *Id.* at 1313.

82. *Id.* at 1313.

83. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2358 (2014).

84. *Id.* at 14. (citing Parker v. Flook, 437 U.S. 593 (1978)).

85. *See, e.g.*, Simon Phipps, *Alice is Killing The Trolls—But Expect Patent Lawyers to Strike Back*, INFOWORLD (Sept. 18, 2014), http://www.infoworld.com/article/2607187/patents/alice-is-killing-the-trolls-but-expect-patent-lawyers-to-strike-back.html.

II. PROFESSOR MARK LEMLEY'S PROPOSAL FOR FUNCTIONAL CLAIMING LIMITATIONS

Professor Lemley[86] has asserted that the answer to many software patentability issues is in stricter application of 35 U.S.C. § 112(f)[87] which would impose restrictions on functional claiming.[88] In advocating for this approach, Lemley argues that an obvious advantage is its ease of implementation, since it would require no rewriting of patent law—rather it would merely require "interpret[ing] the existing statute given the realities of software and modern patent practice."[89] The courts could easily adopt this approach, quickly ending the fractured debates, and resolving the present conflicts surrounding software-related patents.[90]

A. History of Functional Claiming and its Present Use

As detailed by Lemley, the use of functional claiming has varied throughout the years.[91] In the early nineteenth century, "patent claims served as sign posts, not as fence posts," yet toward the middle of the nineteenth century, claim boundaries were greatly expanded and "patentees were using claim language not to signpost what they had done, but to try to define a conceptual area around which they could place legal fence posts."[92] Courts eventually rejected such expansive claiming practices, and although *Halliburton Oil Well Cementing Co. v. Walker*[93] seemingly ended functional claiming as a whole, Congress effectively reinstated its use in Section 112(f) of the Patent Act of 1952—though in the more narrowed form of "means-plus-function" claiming.[94] As noted by Lemley, means-plus-function claim language is currently viewed dismissively as "narrow and easy for potential infringers to evade," so this aspect of the 1952 Act has largely fallen by the wayside.[95]

86. *See supra* note 8.

87. 35 U.S.C. § 112 (2014). The passage of the Leahy-Smith America Invents Act in 2011 changed 35 U.S.C. § 112 into (a)–(f) format. What was once Section 112 paragraph six is now 112(f). *Id.* (as amended Sept. 16, 2011 , P.L. 112-29, § 4(c)).

88. Lemley, *supra* note 26.

89. Mark A. Lemley, *Let's Go Back to Patenting the 'Solution,' Not the 'Problem'*, WIRED (Oct. 31, 2012), www.wired.com/opinion/2012/10/mark-lemley-functional-claiming/. *See also* Lemley, *supra* note 26, at 947–48.

90. Lemley, *supra* note 26, at 947–48 ("The Federal Circuit or the Supreme Court could, with one fell swoop, do away with most of the problem of over-claiming in software patents—and with it, most of the problems with software patents.").

91. *Id.* at 910–19. Infringement inquiries during this time were almost always subject to what is known now as the doctrine of equivalents, and judgments made by courts were done on a case-by-case basis. *Id.* at 910–11.

92. *Id.* at 910–11.

93. Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1 (1946).

94. Lemley, *supra* note 26, at 914–16.

95. *Id.* at 918–19.

Consequently, the practice of pressing boundaries and claiming the outermost periphery of an invention has continued without slowing down.[96]

Since inventors continue to seek and secure broad patent claims,[97] the construction of many claims "are effectively unlimited *as a matter of structure.* The function they perform may be simple or complex, broad or narrow, but in the modern world" many patent claims are written to "effectively cover *any device* that performs that function in any way."[98] These claims go even farther and "assert ownership of any device that is capable of implementing that idea, whether or not the device actually does so."[99] Consequently, software-related patents have essentially reverted to the form that existed before the 1952 Patent Act, without being held to the constraints of Section 112(f).[100]

B. Lemley's Proposal for Eliminating Overbroad Software Patents

Lemley's suggestion for resolving the current state of software patents is very straightforward: "[T]ake seriously the dictate of Section 112(f). . . . [T]ake seriously law that is on the books but doesn't seem to get applied in practice."[101] The language of 35 U.S.C. § 112(f) states:

> *An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.*[102]

In short, claims that use this type of "means" clause define elements by their function rather than their structure—that is, defining elements by what they *do* rather than what they *are*.[103]

Lemley points out that the Federal Circuit has been formalistic in its application of Section 112(f),[104] and it continues to use the presence or absence of the word "means" from a claim to determine which party has the burden of proving that the an element falls under the purview of Section 112(f).[105] However, despite a strict approach for requiring actual algorithms and programing in claims, the realities of

96. *Id.*
97. *Id.* at 919.
98. *Id.* at 920–23 (emphasis original).
99. *Id.*
100. Lemley, *supra* note 26, at 924, 926.
101. *Id.* at 947.
102. 35 U.S.C. § 112(f) (2014).
103. FABER, *supra* note 3.
104. Lemley, *supra* note 26, at, 944–45 n.168.
105. FABER, *supra* note 3, at 3-119.

computer technology have largely been ignored.[106] Where references are made to general purpose computers "programmed to" achieve some function, case law treats those computers as structure and not as a means-plus-function claim with proper limitations.[107] "The 'structure, material or acts' that must support a claim in functional language must be more than mere window-dressing. The intent of [Section 112(f)] was to allow functional claiming only when it was limited to particular implementations of that function, not when it encompassed all feasible ways of achieving the goal."[108]

According to Lemley, it should not make a difference whether a claim is explicitly written in the form "means for doing x," but what matters is whether the claim is "*expressed* as a 'means or step for performing a specified function without the recital of structure, material, or acts in support thereof.'"[109] Abiding by this framework, if a term does not have a generally understood meaning then it should be construed under 112(f) if used in a means–type claim.[110] Examples of such terms include "a device for . . ." or "an element for . . ." or "a mechanism for . . ." and would all be subjected to the more stringent application of Section 112(f).[111] If these claims do not have accompanying structure to sufficiently limit their scope, then those claims should fail on grounds such as indefiniteness, or lack of enablement.

Despite potential obstacles[112] Lemley asserts that faithful application of Section 112(f) may be necessary in order to save many software patents and avoid the tumultuous effects of sweeping invalidation.[113] As Lemley concludes:

> *It is time to end functional claiming (again). Allowing inventors to assert ownership over the problem they solved, rather than merely the way they solved it, is inconsistent with history, with the patent statute, and with good patent policy. It is responsible in large part for the untenable situation software patents have left us in. And while software patent owners may object that they need functional claiming to get effective protection, that objection is unpersuasive, both because of the harm functional claiming causes and because functional patent claims are likely invalid under current law.*

106. *Id*. at 945–46.

107. *Id.* at 946.

108. *Id.*

109. *Id.* at 944 (emphasis added). *See also* FABER, *supra* note 3, at 3-129 (stating that the word "for" in a means-plus-function claim may not need to be used either, and that the claim only be written in means-type form).

110. FABER, *supra* note 3, at 3-129.

111. *Id*. at 3-126.

112. Lemley's thesis does address some of the concerns that others may raise with his proposal. The two major objections addressed are: whether such changes will actually accomplish the desired results, and whether inventors will be unfairly disadvantaged. *See* Lemley, *supra* note 26, at 949–61 (2013).

113. *Id*. at 962–63.

> *A patent should not guarantee insulation from competition. To the contrary, properly understood, patents spur competition by preventing direct imitation while leaving open avenues for alternative development. We have forgotten that lesson in software, to our great cost. Returning to a world in which inventors own their idea, but not the ideas of others, will go a long way towards ensuring that patents encourage rather than retard software innovation.*[114]

III. The Practical Effects of Stricter Functional Claiming Limitations

It is unknown what the actual effects may be from adopting Lemley's proposal until the approach is implemented and tested before judges who understand its proper administration.[115] Yet it is possible to gain some insight into what the consequences may be for software-related patents by applying Lemley's approach to previously litigated matters and carefully observing what differences or effects may result. To accomplish this, several representative patents have been collected and the claims strictly construed under Section 112(f). The analyses demonstrate what types of claims would or would not survive, and what the practical effects may be from adhering to stricter 112(f) application.

A. Claims That Would Not Survive Stricter Applications of Section 112(f)

Greater use of Section 112(f), as advocated by Lemley, would be intended to produce better quality patents.[116] Accordingly, it should be expected that many patents would therefore be invalidated since they would not be of sufficient quality to satisfy the higher standards. An example of claims which would not survive this type of scrutiny comes from U.S. Patent 5,970,479 (the '479 patent) at issue in *Alice Corp. v. CLS Bank*.[117] Before the court were claims 33 and 34, which read:

> *33. A method of exchanging obligations as between parties, each party holding a credit record and a debit record with an exchange institution, the credit records and debit records for exchange of predetermined obligations, the method comprising the steps of:*
>
> *(a) creating a shadow credit record and a shadow debit record for each stakeholder party to be held independently by a supervisory institution from the exchange institutions;*

114. *Id*. at 964.
115. *See supra* note 11 and accompanying text.
116. *See supra* Part II.B.
117. 134 S. Ct. 734 (U.S. 2013).

> *(b) obtaining from each exchange institution a start-of-day balance for each shadow credit record and shadow debit record;*
>
> *(c) for every transaction resulting in an exchange obligation, the supervisory institution adjusting each respective party's shadow credit record or shadow debit record, allowing only these transactions that do not result in the value of the shadow debit record being less than the value of the shadow credit record at any time, each said adjustment taking place in chronological order; and*
>
> *(d) at the end-of-day, the supervisory institution instructing ones of the exchange institutions to exchange credits or debits to the credit record and debit record of the respective parties in accordance with the adjustments of the said permitted transactions, the credits and debits being irrevocable, time invariant obligations placed on the exchange institutions.*
>
> *34. The method as in claim 33, wherein the end-of-day instructions represent credits and debits netted throughout the day for each party in respect of all the transactions of that day.*[118]

The '479 patent would be construed under 112(f) because use of the language "a method for . . ." in this context is a means clause. In order to survive, the claim requires limiting structure, yet there is none present. Terms with generally understood meanings such as "credit record" and "debit record" simply describe features which do not perform a tangible function. Failure to show structure that performs the function means the claims are invalid,[119] which would effectively limit the scope of the '479 patent. If allowed to survive, these claims would extend to anything that performs the function—not just computers—and could be asserted in absurd situations against devices such as an abacus, graphing calculators, or even handwritten record books.

The '479 patent is an example of how stricter application of Section 112(f) could be beneficial. An obviously bad patent—which was nevertheless granted by the U.S. Patent and Trademark Office—would not survive the application of greater functional claiming limitations. Instead of creating a monopoly over the foundational principles of escrow, the claimant would be forced to establish that the patent had something with a more "inventive" quality to it.[120] Under this type of analysis, it would be easier to discern the presence or absence of that elusive

118. U.S. Patent No. 5,790,479 col. 65, l.23–53 (filed Aug, 4, 1998).

119. *See, e.g.*, *In re* Donaldson Co., 16 F.3d 1189 (Fed. Cir. 1994) (en banc); *see also* Crouch, *supra* note 3.

120. *See, e.g.*, CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1273-92 (Fed. Cir. 2013) (noting that patentability requires an "inventive concept" that provides "something more" to a viable claim) (per curiam) (Lourie, J., concurring).

characteristic "something more"—which the courts have inadequately articulated—and could resolve many of the difficulties in trying to express what makes a patent worthy of protection.

Another example of claims which would not survive can be seen in the case of *Parker v. Flook*.[121] The patent application concerned a "Method for Updating Alarm Limits" and contained the following claim:

> *1. A method for updating the value of at least one alarm limit on at least one process variable involved in a process comprising the catalytic chemical conversion of hydrocarbons wherein said alarm limit has a current value of*
>
> *Bo + K*
>
> *wherein Bo is the current alarm base and K is a predetermined alarm offset which comprises:*
>
> *(1) Determining the present value of said process variable, said present value being defined as PVL:*
>
> *(2) determining a new alarm base B1, using the following equation:*
>
> *B1 = Bo (1.0 - F) + PVL (F)*
>
> *where F is a predetermined number greater than zero and less than 1.0;*
>
> *(3) determining an updated alarm limit value which is defined as*
>
> *B1 + K; and, thereafter*
>
> *(4) adjusting said alarm limit to said updated alarm limit value.*[122]

Although the Supreme Court decided that the claims in *Flook* were directed to unpatentable subject matter,[123] they would also fail under 112(f) considerations because they are not tied to any structure, and claim unlimited use of a particular function. Reading the various elements, there is nothing present which demonstrates structure that performs the function—only a description of the function itself.

121. 437 U.S. 584 (1978).
122. Application of Flook, 559 F.2d 21 (C.C.P.A. 1977).
123. *Flook*, 437 U.S. at 594.

Further examples of patents which would not pass strict 112(f) application include the various claims at issue in *Bancorp Services, LLC v. Sun Life Assurance Co. of Canada*.,[124] *CyberSource Corp. v. Retail Decisions, Inc.*,[125] *Dealertrack, Inc. v. Huber*,[126] *Netgear, Inc. v. Ruckus Wireless, Inc.*,[127] *Noah Systems, Inc. v. Intuit, Inc.*,[128]

124. Bancorp Servs., L.L.C. v. Sun Life Assur. Co. of Canada *(U.S.)*, 687 F.3d 1266 (Fed. Cir. 2012) (claims of U.S. Patents 5,926,792 and 7,249,037). *See, e.g.*, claim 1 of U.S. Patent No. 5,926, 792, which reads:

> A computer system for managing a life insurance policy on behalf of a policy holder, the computer system comprising:
>
> generating means for generating a life insurance policy including a stable value protected investment with an initial value based on a value of underlying securities;
>
> fee calculating means for calculating fee units for members of a management group which manage the life insurance policy;
>
> credit calculating means for calculating surrender value protected investment credits for the life insurance policy;
>
> investment determining means for determining an investment value and a value of the underlying securities for the current day;
>
> policy calculating means for calculating a policy value and a policy unit value for the current day;
>
> storing means for storing the policy unit value for the current day; and
>
> one of:
>
> removing means for removing the fees units for members of the management group which manages the life insurance policy, and
>
> accumulating means for accumulating fee units on behalf of the management group.

125. CyberSource Corp. v. Retail Decisions, Inc., 654 F.3d 1366 (Fed. Cir. 2011) (claims 2 and 3 of U.S. Patent No. 6,029,154 (filed Jul. 28, 1997)).

126. Dealertrack, Inc. v. Huber, 674 F.3d 1315 (Fed. Cir. 2012) (claims 7–9, 12, 14, 16, and 17 of U.S. Patent No. 6,587,841 and claims 1, 3, and 4 of U.S. Patent No. 7,181,427). *See, e.g.*, claims 1 and 2 of U.S. Patent No. 6,587,841 (filed Jun. 30, 1998), which read:

> 1. A credit application and routing system including a central processor having and executing a program and further comprising:
>
> data input means for selectively receiving credit application data from respective applicants at remote locations; and
>
> routing means for selectively forwarding the credit application data to remote funding sources
>
> and forwarding funding decision data from the funding sources to the respective applicants, wherein the routing means comprises means for sending at least a portion of a credit application to more than one of said remote funding sources sequentially until a funding source returns a positive funding decision or until all funding sources have been exhausted.

State Street Bank & Trust Co. v. Signature Financial Group, Inc.,[129] and *Ultramercial, Inc. v. Hulu, LLC*.[130]

Not only would strict Section 112(f) application provide a finer screen for patent quality, but the *Flook* patent and others reveal the potential for additional benefits from this approach. Notably, use of Section 112(f) would provide courts with greater flexibility in decision making rather than relying so much on the coarse filter of Section 101.[131] This would add an arrow to the judicial quiver—one founded in purely statutory considerations—and avoid the need for courts to try and formulate an effective rule. Lemley's approach fits this tactic of applying law that is already on the books rather than making unnecessary categorical changes.[132] As a result, stricter application of Section 112(f) could make the role of the courts

> 2. The credit application and routing system according to claim 1, wherein said central processor further provides at least one of a loan calculator, lender product news and funding source articles.

127. Netgear, Inc. v. Ruckus Wireless, Inc., 852 F. Supp. 2d 470, 471–72 (D. Del. 2012) (claims of U.S. Patent Nos. 5,812,531; 6,621,454; 7,263,143; and 5,507,035).

128. Noah Sys., Inc. v. Intuit Inc., 675 F.3d 1302 (Fed. Cir. 2012) (claims 12–17, 29–38, and 40–56 of U.S. Patent No. 5,875,435).

129. State St. Bank & Trust Co. v. Signature Fin. Grp., Inc., 149 F.3d 1368 (Fed. Cir. 1998) (claims of U.S. Patent No. 5,193,056). *See, e.g.*, claim 1 of U.S. Patent No. 5,193,056 which reads:

> 1. A data processing system for managing a financial services configuration of a portfolio established as a partnership, each partner being one of a plurality of funds, comprising:
>
> (a) computer processor means for processing data;
>
> (b) storage means for storing data on a storage medium;
>
> (c) first means for initializing the storage medium;
>
> (d) second means for processing data regarding assets in the portfolio and each of the funds from a previous day and data regarding increases or decreases in each of the funds, assets and for allocating the percentage share that each fund holds in the portfolio;
>
> (e) third means for processing data regarding daily incremental income, expenses, and net realized gain or loss for the portfolio and for allocating such data among each fund;
>
> (f) fourth means for processing data regarding daily net unrealized gain or loss for the portfolio and for allocating such data among each fund; and
>
> (g) fifth means for processing data regarding aggregate year-end income, expenses, and capital gain or loss for the portfolio and each of the funds.

130. Ultramercial, Inc. v. Hulu, LLC, 722 F.3d 1335 (Fed. Cir. 2013) (claims of U.S. Patent No. 7,346,545).

131. 35 U.S.C. § 101 (2014). ("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."). *Id.*

132. *See supra* note 101.

easier by providing more distinct guidelines to consider when making decisions on complicated and highly technical patents.

B. Claims That Would Survive Stricter Applications of Section 112(f)

Although low-quality patents would easily be screened out under Lemley's proposal, the standard cannot also be so stringent that it causes high-quality patents to be rejected as well. If patents that are deserving of protection cannot survive, then the patent system will fail to provide the incentives that form part of its foundation.[133] Fortunately, Lemley's approach does not appear to set the bar too high.[134] An example of a patent which would pass strict Section 112(f) application is seen in *RLIS v. Allscripts Healthcare Solutions*.[135] At issue were claims 2, 4, and 5 of U.S. Patent 5,823,948 (the '948 patent), which read:

> *2. A method of patient record documentation, tracking and order entry, comprising providing software in file servers, providing software from the file servers through a network hub and network to multiple peripheral CPUs, inputting data on data entry screens on monitors connected to the CPUs, the inputting comprising entering patient data in the multiple CPUs by touch screens, mouses and keyboards in response to the data entry screens on monitors connected to the CPUs, transferring the patient data from the CPUs to the file servers, making a record for individual patients, dictating portions of the record that are unique to particular patients, transmitting the dictation over lines to a transcription center, transcribing the dictation and transmitting the dictation transcriptions to at least one communication server, feeding the dictation transcriptions to the file servers as text, storing the text with the patient data on particular patients in the file servers, storing word and sentence generation and coordination software in the peripheral CPUs, generating text sentences in medical English text from patient data, displaying on the monitors connected to the peripheral CPUs the text sentences in medical English text combined with the text from the dictation transcriptions assembled as text summaries, and providing the text summaries from the peripheral CPUs to printers via the network for generation of printed patient textual reports.*
>
> *4. The method of claim 2, further comprising producing nurses' notes data by entering the patient data with touch screens, mouses and keyboards, transferring the nurses' notes data from the peripheral CPUs to the file*

133. *See, e.g.*, *supra* note 27 and accompanying text.

134. *Supra* Part II.B.

135. *RLIS, Inc. v. Allscripts Healthcare Solutions, Inc.*, 3:12-CV-208, 2013 WL 3772472 (S.D. Tex. July 16, 2013) (U.S. Patent Nos. 7,076,436, and 5,823,948).

> *servers, storing the nurses' notes data in the file servers, transmitting the nurses' notes data to the peripheral CPUs, recompiling the nurses' notes data into nurses' notes text in the CPUs, displaying the nurses' notes text on peripheral CPU monitor screens, and printing the nurses' notes text on the printer.*
>
> *5. The method of claim 2, further comprising storing prephrased text examples in the peripheral CPUs, and preliminarily inputting prephrased personalized text by individual physicians and nurses, and compiling the personalized text with the patient data for producing medical English text summaries and reports.*[136]

In this patent, the language "a method for . . ." indicates a means clause which would bring these claims under 112(f). Although the method described is seemingly vague, the language in the '948 patent is tied to several elements of structure—unlike the language in either *Alice v. CLS* or *Parker v. Flook*.[137] For instance, the initial method claim in the '948 patent is supported by "file servers" and "peripheral CPUs" as structure. Or looking at language such as "inputting data on data entry screens on monitors connected to the CPUs, the inputting comprising entering patient data in the multiple CPUs by touch screens, mouses and keyboards," the description of "inputting data" would be a function, yet it is grounded in supporting elements of "CPUs" and "touch screens, mouses and keyboards."[138] Another example is "transferring the patient data from the CPUs to the file servers," where transferring patient data is a function but it is also supported by the physical file servers and CPUs.[139] There are many other instances of functions in the '948 patent, including transcribing and transmitting dictation, generating and displaying text summaries, and producing and transmitting notes data. Yet all of these functions are tied to structural elements including CPUs, communication servers, file servers, monitors, printers, touch screens, and peripheral CPUs. This patent is an excellent example of how Lemley's proposal would be an easily administrable method for distinguishing the quality of claims.

Another example of a patent which would survive stricter application of 112(f) can be found in the case of *Ibormeith v. Mercedes-Benz*.[140] At issue was U.S. Patent 6,313,749 (the'749 patent), which read in part:

> *1. A sleepiness monitor for a vehicle driver, or machine operator, comprising:*

136. U.S. Patent No. 5,823,948 (filed July 8, 1996).
137. *See supra* Part III.A.
138. U.S. Patent No. 5,823,948 (filed July 8, 1996).
139. *Id.*
140. Ibormeith IP, LLC v. Mercedes-Benz USA, LLC, 732 F.3d 1376 (Fed. Cir. 2013).

a sensor for sensing a driver or operator control input;

a memory for storing an operational model that includes a physiological reference model of driver or operator circadian rhythm pattern(s) and a vehicle or machine operating model or algorithm;

computational means for weighting the operational model according to time of day in relation to the driver or operator circadian rhythm pattern(s) and for deriving, from the weighted model, driver or operator sleepiness condition and producing an output determined thereby; and

a warning indicator triggered by the computational means output, to provide a warning indicator of driver or operator sleepiness.

2. The sleepiness monitor as claimed in claim 1, including a driver personal data entry interface, for entry of driver sleep pattern, age, sex, and recent alcohol consumption.

3. The sleepiness monitor as claimed in claim 1, including provision, by way of switches, for input of responses to predetermined questions upon driver or operator condition, including recent sleep history.

4. The sleepiness monitor as claimed in claim 1, wherein the sensor comprises a magnetic flux coupled, inductive sensor for rate of change of vehicle or machine steerage.

5. The sleepiness monitor as claimed in claim 1, including a further sensor for vehicle acceleration and/or speed.

6. The sleepiness monitor as claimed in claim 1, including a further sensor for vehicle cab temperature.

7. The sleepiness monitor as claimed in claim 1, including a further sensor for ambient light.

8. A vehicle or machine incorporating a sleepiness monitor as claimed in claim 1.[141]

Although the patent appears to claim only a physical device for monitoring sleepiness, it also reads as claiming the *function* of monitoring sleepiness. In

141. U.S. Patent No. 6,313,749 col. 16 (filed Jan. 5, 1998).

particular, claim 1 includes the language "computational means for . . ." which is used in the style of a means-type limitation that would require the claims to be construed under 112(f).[142] The "computational means" language is troublesome for claim 1 since there is no apparent generally understood meaning for the term. Moreover, claim 1 does not describe sufficient structure since terms such as "a sensor for sensing" or "a memory for storing" or "a warning indicator" are all too general to offer adequate limitation.[143] However, dependent claims 2 through 8 provide the structural limitations which would allow claim 1 to survive.[144] For instance, claim 4 describes the sensor in greater detail, adding that it is "a magnetic flux coupled, inductive sensor."[145] Although the other claims do not provide particularly concrete language for limiting the scope of the claims, the Ex Parte Reexamination Certificate issued for this patent provides a stronger basis for its survival under Section 112(f). Specifically, claims 10 and 11 of the Certificate eliminate the term "computational means for" and use more detailed language, such as "an internal microprocessor specially programmed to . . . ." This language and other descriptions in claims 10 and 11 appear to provide the adequate limitations that should be present in high quality patents. [146]

Looking at examples such as the '948 and the '749 patents demonstrates that Lemley's proposed approach to software-related patents is a reasonable one. Rather

142. *Id.*

143. *Id.*

144. Claim 9 of the '749 patent would likely be invalidated under this type of analysis because it does not have similar language to claim 1 that would effectively limit the "computational means for" term. *See* U.S. Patent No. 6,313,749, col. 16 (filed Jan. 5, 1998).

145. *Id.*

146. *See, e.g.*, claim 10 of the '749 patent which reads:

> 10. A sleepiness monitor for a vehicle driver, or machine operator, comprising:
>
> a sensor for sensing a driver or operator control input, wherein said sensor is for sensing steering transitions about a reference position;
>
> a memory for storing an operational model that includes a pre-loaded physiological reference model of a driver or operator circadian rhythm pattern(s) and a vehicle or machine operating model or algorithm;
>
> an internal microprocessor specially programmed to incorporate said sensed steering transitions into said vehicle or machine operating model or algorithm and to weight said operational model according to time of day in relation to the driver or operator circadian rhythm pattern(s), derive, from the weighted model, a driver or operator sleepiness condition and produce an output determined thereby; and
>
> a warning indicator, triggered by the processor output, to provide a warning indication of driver or operator sleepiness, wherein said warning indicator includes a visual warning shown on or through a display screen.

Ex Parte Reexamination Certificate, U.S. Patent No. 6,313,749.

than create an insurmountable hurdle, claims are merely subject to a higher standard—one that has a statutory basis—which imposes more realistic limitations and encourages better quality of claims.[147]

IV. CONCLUSION

Adopting stricter application of Section 112(f) would be an effective and consistent approach to resolving many of the problematic issues with software-related patents. Such a standard would make it easier for entities to navigate the enormous size and complexity of the present system, or provide tools to hack through dense patent thickets and create more equitable bargaining positions with other companies and NPEs.[148] It should put an end to the fractured debates on how to "solve" problems with software-related patents,[149] and relieve the courts from formulating inadequate judicial frameworks.[150] Applying strict 112(f) limitations would also ensure that the bar for protection is not set so high that it would obstruct innovative ideas from receiving due protection.[151] This approach is not a short cut which would make for long delays, nor should there be dire concern about "the death of hundreds of

147. An example of how Section 112(f) could encourage better quality claiming can be seen in the patents at issue in Research Corp. Technologies, Inc. v. Microsoft Corp. 627 F.3d 859 (Fed. Cir. 2010). Looking at the six patents in question, the language used would not pass the strict application of 112(f) that Lemley advocates for. However, like the *Ex Parte* Reexamination Certificate in *Ibormeith*, *supra* note 146, small changes to the language would allow these patents to pass a more stringent standard. For example, claim 1 of U.S. Patent No. 5,477,305 reads:

> 1. Apparatus for performing the halftoning of a gray scale image by utilizing a pixel-by-pixel comparison of said image against a blue noise mask array, comprising:
>
> a) a first memory for storing said blue noise mask array;
>
> b) a scanner for scanning said gray scale image to be halftoned to create a gray scale image array on a pixel-by-pixel basis;
>
> c) a second memory for storing said gray scale image array;
>
> d) a comparator for comparing, on a pixel-by-pixel basis, the value of each corresponding pixel in said blue noise mask array and said gray scale image array to produce a binary image array; and
>
> e) a converter for converting said binary image array to the desired halftoned image.

Language such as "a first memory" or "a scanner" or "a second memory" or "a comparator" or "a converter" do not sufficiently describe limiting structure. Yet with simple changes—such as making use of more generally understood terms and more specific language—these claims would most likely survive the application of higher standards.

148. *See supra* Part I.A.

149. *See supra* Part I.B.

150. *See supra* Part I.C.

151. *See supra* Part II.B, III.

thousands of patents,"[152] even though numerous claims would not survive. The examples in Part III demonstrate exactly the kind of low-quality patents which should have been screened out by the U.S. Patent and Trademark Office in the first place,[153] while also demonstrating how small changes can increase the overall quality of patent claims.[154] Given the increasing volume and rate of software-related patents involved in litigation,[155] higher standards must be implemented to lessen the daunting situation that innovators presently face.[156] Raising and enforcing higher standards should not only increase patent quality, but also facilitate greater innovation such that the world may grow more enlightened and ensure that knowledge may be more equally diffused.

152. CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1313 (Fed. Cir. 2013) (per curiam).
153. *See supra* Part III.A.
154. *See supra* Part III.B.
155. *See supra* notes 3– 4, and accompanying text.
156. *See supra* notes 20 and 31.