UNC SCHOOL OF LAW

NORTH CAROLINA JOURNAL OF LAW & TECHNOLOGY

Volume 20 | Issue 2 Article 3

12-1-2018

Subject Matter Eligibility and Functional Claiming in Software Patents

Michael Xun Liu

Follow this and additional works at: https://scholarship.law.unc.edu/ncjolt

Part of the Law Commons

Recommended Citation

Michael X. Liu, *Subject Matter Eligibility and Functional Claiming in Software Patents*, 20 N.C. J.L. & TECH. 227 (2018).
Available at: https://scholarship.law.unc.edu/ncjolt/vol20/iss2/3

This Article is brought to you for free and open access by Carolina Law Scholarship Repository. It has been accepted for inclusion in North Carolina Journal of Law & Technology by an authorized editor of Carolina Law Scholarship Repository. For more information, please contact law_repository@unc.edu.

NORTH CAROLINA JOURNAL OF LAW & TECHNOLOGY
VOLUME 20, ISSUE 2: DECEMBER 2018

SUBJECT MATTER ELIGIBILITY AND FUNCTIONAL CLAIMING IN SOFTWARE PATENTS

Michael Xun Liu

Subject matter eligibility and functional claiming are considered separate doctrines in patent law. Conceptually, subject matter eligibility relates to the types of ideas that can be patented, whereas limits on functional claiming constrain how patentees can claim their inventions. In practice, however, patents that recite functional claims are also more likely to be invalidated for failing to recite patentable subject matter. This trend is especially prevalent in the software field, where courts often hold the function or end result of a computer program represents an unpatentable abstract idea.

Critics argue this judicial approach to software patents improperly conflates "what is patentable" with "how patentees can claim their inventions." To rein in this practice, Congress has introduced legislation that would narrow judicial authority over patentable subject matter. The current legislative proposals, however, do not address underlying policy issues surrounding functional claims in software patents. They will also prevent courts from invalidating patents that broadly preempt future innovation. Instead of limiting judicial authority over patentable subject matter, policymakers should address functional software claims more directly, such as adopting a more flexible interpretation of means-plus-function claims under 35 U.S.C. § 112(f) or delegating rulemaking authority to the Patent and Trademark Office ("PTO").

I. INTRODUCTION

Modern software can perform functions that few could imagine in previous decades. A typical smartphone can run applications that navigate drivers, pay for coffee, hail a rideshare, and even diagnose diseases. Programmers should certainly have incentives to dream up new ways to make life easier and work more productive. The hard question for policymakers, however, is whether those incentives should be in the form of patent rights and, if so, how extensive those rights should be. In a world that depends on software at every turn, it may seem inconceivable that basic questions about software patentability remain unsettled. Yet the extent to which software is patentable and the permissible scope of software claims are both still hotly debated. While these questions have never been definitively resolved, the Supreme Court's recent decision in *Alice Corp. v. CLS Bank International* brought them squarely back into the limelight.[1] The ensuing controversy has generated renewed calls for a legislative solution to software patents.[2]

[1] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2360 (2014); *see* Jeffrey A. Lefstin et al., *Final Report of the Berkeley Center for Law & Technology Section 101 Workshop: Addressing Patent Eligibility Challenges*, BERKELEY TECH. L. J. (forthcoming 2018) (manuscript at 555), https://ssrn.com/abstract=3050093.

[2] *See infra* Section V.

To understand why software patents pose unique challenges for courts and legislators, it helps to start with a few foundational concepts. Congress is constitutionally empowered to "promote the progress of science and useful arts, by securing . . . inventors the exclusive right to their respective writings and discoveries."[3] But that does not mean all scientific advancements are patentable or inventors own every aspect of their discovery. Courts and policymakers have long recognized some patents do little to promote scientific or technological progress and may even hinder subsequent innovation.

First, most would agree that limits should exist for patentable subject matter, which are the types of discoveries eligible for patenting. For instance, natural laws and abstract ideas cannot be patented.[4] Thus, even the discoverer of groundbreaking ideas like Bernoulli's principle or the theory of relativity cannot claim the exclusive use of those concepts.[5] Second, even if the invention falls within the realm of patentable subject matter, the law further constrains *the scope* of the inventor's resulting patent right.[6] Ideally, the inventor's exclusive right should be commensurate with their contribution to technology.[7] Historically, one way courts limited patent scope is by prohibiting attempts to claim the *function* of a device or process.[8] The rationale behind this prohibition is that functional claims preclude others from developing new and different ways of performing the same function.[9]

To illustrate these limits on patent eligibility and scope, consider a hypothetical patent on an airplane. According to Bernoulli's principle, fast moving fluids exert lower pressure than slower moving fluids.[10] Using this principle, an airplane generates lift by

[3] U.S. CONST. art. I, § 8, cl. 8.

[4] Diamond v. Diehr, 450 U.S. 175, 185 (1981).

[5] Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980).

[6] Wyeth v. Stone, 30 F. Cas. 723, 727 (C.C.D. Mass. 1840) ("A claim broader than the actual invention of the patentee is, for that very reason, upon the principles of the common law, utterly void, and the patent is a nullity.").

[7] *Id.*

[8] Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 10 (1946).

[9] *Id.*

[10] BERNOULLI'S PRINCIPLE, NAT'L AERONAUTICS & SPACE ADMIN. 4 (2010), https://www.nasa.gov/sites/ default/files/atoms/files/bernoulli_principle_k-4.pdf.

forcing the air above its wing to travel faster than the air below its wing.[11] In this hypothetical, the inventor is the first person to learn how to harness Bernoulli's principle to generate lift, and she invents a working propeller plane based on this understanding. Bernoulli's principle is a natural law that cannot be patented.[12] An airplane, however, is a machine that applies the natural law and therefore falls within the realm of patent-eligible subject matter.[13]

Even though an airplane is patent-eligible, there are still limits on *how* the inventor can claim her invention. For instance, our hypothetical inventor might claim "a machine with a fuselage, fixed wings, and a propeller" arranged in a specific way. This is an accurate description of her invention because it covers the design (i.e., a propeller plane) she created. Alternatively, the inventor might claim "a machine with a fuselage and a means for applying Bernoulli's principle to fly." From a technical and linguistic perspective, this second claim is *also* an accurate description of her invention. The difference, however, is the second claim is directed to the function of an airplane, but not *how* it performs the function. As a result, the literal scope of the second claim extends beyond propeller planes or their obvious variants.[14] Instead, the claim language also covers totally different machines that apply Bernoulli's principle to fly, such as helicopters.

In theory, subject matter eligibility addresses *what* ideas can be patented, and limits on functional claiming constrain *how* those ideas can be patented. These issues are governed by separate statutory provisions. In addressing subject matter eligibility, 35 U.S.C. § 101 states any "new or useful process, machine, manufacture, or composition of matter" is patentable.[15] By contrast, 35 U.S.C. § 112(f) limits the scope of functional claims.[16] Under § 112(f), if a patentee drafts a claim "as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof," the claim will be construed to

[11] *Id.* at 5.

[12] Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980).

[13] Le Roy v. Tatham, 55 U.S. 156, 175 (1852).

[14] Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 10 (1946).

[15] 35 U.S.C. § 101 (2012).

[16] *Id.* § 112(f).

cover only what is disclosed in the specifications.[17] That way, if a patent describes a propeller plane but claims "a machine with a fuselage and a means for applying Bernoulli's principle to fly," § 112(f) will limit the claim to propeller planes and their equivalents.[18]

In reality, the distinction between "what is patentable?" and "how can it be patented?" is not always so clear. We might say a patent on "a machine that applies Bernoulli's principle to fly" recites a patent-eligible idea because it is directed to an airplane but claims the idea in an impermissible manner because it is too broad. Some might also argue the claim is too abstract to be patentable.[19] That is because it covers an idea for how Bernoulli's principle can be used without reciting any technical aspects of the machine's design or components.[20] In other words, the claim is directed to the idea of using a natural law in the technological environment of a "flying machine," but it does not cover any specific technology for actually implementing the idea.[21]

Accordingly, the same claim might be abstract or functional depending on how we frame the analysis. This distinction, however, has real consequences. If a claim is abstract, then it is invalid.[22] By contrast, if the claim is non-abstract but functional, then its scope is

[17] *Id.*

[18] As the Federal Circuit explained in *Williamson v. Citrix Online, LLC*:

> Congress struck a balance in allowing patentees to express a claim limitation by reciting a function to be performed rather than by reciting structure for performing that function, while placing specific constraints on how such a limitation is to be construed, namely, by restricting the scope of coverage to only the structure, materials, or acts described in the specification as corresponding to the claimed function and equivalents thereof.

Williamson v. Citrix Online, LLC, 792 F.3d 1339, 1347 (Fed. Cir. 2015).

[19] Elec. Power Grp., LLC v. Alstom S.A., 830 F.3d 1350, 1356 (Fed. Cir. 2016) ("[E]ssentially result–focused, functional character of claim language has been a frequent feature of claims held ineligible under § 101 . . . .").

[20] *Id.*

[21] *Id.* at 1354 ("[L]imiting the claims to the particular technological environment of power–grid monitoring is, without more, insufficient to transform them into patent–eligible applications of the abstract idea at their core.").

[22] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2360 (2014).

limited to the "corresponding structure, material, or acts described in the specification and equivalents thereof."[23]

This distinction is particularly hard to make for software patents. On a fundamental level, software can be characterized as a set of instructions or algorithms, which the Supreme Court has held to be abstract.[24] Nevertheless, software is always embodied in some physical form, whether it is stored in a computer readable medium (i.e. a hard-drive) or running on an electronic device.[25] As a result, patentees often try to claim the implementation of software on physical devices.[26] For example, a typical software patent might recite "a computer readable medium containing program instructions" to perform a set of functions.[27] Although this claim is technically directed to a physical device, one could argue taking a set of instructions and saying "apply it on a computer" is no less abstract than the instructions themselves.

At the same time, software patents also raise functional claiming issues. Software claims are rarely limited to specific code and are usually directed to the software's higher-level functions.[28] Such claims can preclude other programmers from writing different code to perform the same function.[29] Often, software patents are asserted against later-developed programs that perform the claimed function but operate in fundamentally different ways, thus raising concerns that software patents hinder innovation or serve as tools for extorting businesses through litigation.[30]

[23] 35 U.S.C. § 112(f) (2011); *see* Williamson, 792 F.3d at 1347.

[24] Kevin Emerson Collins, *Patent Law's Functionality Malfunction and the Problem of Overbroad, Functional Software Patents*, 90 WASH. U. L. REV. 1399, 1455–56 (2013); Gottschalk v. Benson, 409 U.S. 63, 64 (1972).

[25] *Alice*, 134 S. Ct. at 2360 ("Nearly every computer will include a 'communications controller' and 'data storage unit' capable of performing the basic calculation, storage, and transmission functions required by the method claims.").

[26] Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 WIS. L. REV. 905, 919–22.

[27] *Id*. at 920.

[28] *Id.* at 919–22.

[29] *See id.* at 923.

[30] *See* James Bessen, *The Patent Troll Crisis is Really a Software Patent Crisis*, WASH. POST (Sep. 3, 2013), https://www.washingtonpost.com/news/the–

Despite uncertainty surrounding their validity and scope, the Patent and Trademark Office (PTO) took a liberal approach to software patents throughout the 1990s and early 2000s.[31] Within that time, the agency also issued many patents with functional claim language.[32] By 2011, the Government Accountability Office (GAO) estimated over half of all issued patents were related to software.[33] Moreover, software patents accounted for the majority of patent lawsuit filings.[34] Around the same time, another study concluded most litigated patents used some form of functional claiming.[35]

This all changed in 2014 with the Supreme Court's decision in *Alice v. CLS Bank*,[36] which dialed back software patents significantly.[37] In *Alice*, the Court held a computer-implemented method of mitigating financial risk was not patent-eligible even though the claims included hardware components.[38] After *Alice*, lower courts frequently invalidated functional software claims under § 101.[39] The Federal Circuit observed "the essentially result-focused, functional character of claim language has been a frequent feature of claims held ineligible under § 101."[40] In another decision, the court referenced the "foundational patent law principle[] that a result, even an innovative result, is not itself patentable."[41] Some judges and practitioners, however, criticize these decisions for

switch/wp/2013/09/03/the–patent–troll–crisis–is–really–a–software–patent–crisis/?utm_term=.8909c5855d14.

[31] *See* U.S. GOV'T ACCOUNTABILITY OFFICE, GAO-13-465, INTELLECTUAL PROPERTY: ASSESSING FACTORS THAT AFFECT PATENT INFRINGEMENT LITIGATION COULD HELP IMPROVE PATENT QUALITY 13 (2013).

[32] *See* Colleen V. Chien & Aashish R. Karkhanis, *Functional Claiming and Software Patents*, 40–41 (Santa Clara Univ. Sch. L., Working Paper No. 06-13, 2013), https://ssrn.com/abstract=2215867.

[33] *Id.*

[34] *Id.*; Keith N. Hylton, *Patent Uncertainty: Toward a Framework with Applications*, 96 B.U. L. REV. 1117, 1126 (2016).

[35] *See* Chien & Karkhanis, *supra* note 32, at 40–41.

[36] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014).

[37] *See id.* at 2360.

[38] *Id.*

[39] Paul R. Gugliuzza & Mark A. Lemley, *Can a Court Case Change the Law by Saying Nothing?*, 71 VAND. L. REV. 765, 774 (2018).

[40] Elec. Power Grp., LLC v. Alstom S.A., 830 F.3d 1350, 1356 (Fed. Cir. 2016).

[41] Finjan, Inc. v. Blue Coat Sys., Inc., 879 F.3d 1299, 1305 (Fed. Cir. 2018).

conflating distinct inquiries under § 101 and § 112.[42] And Congress has introduced legislative amendments that would prohibit courts from considering any § 112 issues in the patent eligibility analysis.[43]

Although *Alice* brought judicial and Congressional attention to software patents, the current debate over § 101 also reflects decades-long concerns about the nature of software and how it should be treated under patent law. In this Article, I show how judicial concerns about functional software claims underpin the post-*Alice* approach to software patent eligibility. I argue that differing opinions on functional claiming help explain inconsistencies in the Federal Circuit's recent § 101 decisions. In particular, courts have held that software functions are inherently abstract and ineligible.[44] By contrast, others reason software functions are not abstract per se and will look beyond the claim language to evaluate whether the claimed software functions are tied to technological improvements.[45]

I also explain why legislative proposals to amend § 101 should address long-standing issues relating to functional software claims. The current legislative proposals, however, largely fail to do so. Instead, the leading proposals would prohibit courts and the PTO from rejecting broad, functional software patents without resolving underlying concerns about such patents, including their unclear scope and potential to preempt subsequent innovation. I also identify one proposal that, despite its flaws, has the potential to improve

[42] *See* Amdocs (Israel) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1302 (Fed. Cir. 2016); AM. INTELLECTUAL PROP. LAW ASS'N., AIPLA LEGISLATIVE PROPOSAL AND REPORT ON PATENT ELIGIBLE SUBJECT MATTER 3–4 (2017) [hereinafter AIPLA PROPOSAL], https://www.aipla.org/docs/default-source/uploadedfiles/documents/resources2/reports/2017aipladirect/documents/aipla-report-on-101-reform-5-19-17-errata.pdf?sfvrsn=138c9ce7_1.

[43] *See* Restoring America's Leadership in Innovation Act of 2018, H.R. 6264, 115th Cong. § 7(c) (2d Sess. 2018); see also *infra* Section V.

[44] *See, e.g.*, Affinity Labs of Tex., LLC v. Amazon.com Inc., 838 F.3d 1266, 1269 (Fed. Cir. 2016); Apple, Inc. v. Ameranth, Inc., 842 F.3d 1229, 1241 (Fed. Cir. 2016).

[45] *See Amdocs*, 841 F.3d at 1295; Visual Memory LLC v. NVIDIA Corp., 867 F.3d 1253, 1256 (Fed. Cir. 2017).

uniformity and certainty for software patents.[46] This proposal would require courts to analyze functional software claims under § 112(f)—or a new provision to similar effect—instead of invalidating the claims altogether.[47] The caveat is it will require guidelines on what types of disclosures are sufficient to support functional software claiming. Institutionally, the PTO is likely better suited to establish such guidelines.

This Article proceeds in five sections. Section II provides an overview of the evolution of judicial limits on subject matter eligibility and functional claiming through nineteenth century judicial decisions. It also explains why the 1952 Patent Act included a legislative compromise for functional claims. Section III shows how software patents create conceptual difficulties for both patent eligibility and functional claiming analysis. This section also reviews how Congress and the courts tried—and failed—to develop a uniform framework for analyzing the patentability of software. Section IV discusses the judicial approach to software patents after the Supreme Court's decision in *Alice*. It shows how differing views on the nature of software functions underpin the post-*Alice* approach to software patent eligibility. Finally, Section V discusses the implications for the proposed legislative reform of § 101.

II. BACKGROUND ON SUBJECT MATTER ELIGIBILITY AND FUNCTIONAL CLAIMING

The exclusive right of an inventor "depends entirely upon the provisions of the acts of Congress."[48] Perhaps counterintuitively, the statutes provide little guidance on what is patentable.[49] Congress has

[46] *See* Benjamin C. Stasa & David C. Berry, *Fixing Patent Eligibility by Limiting Scope to Disclosed Embodiments*, PATENTLYO (June 4, 2018), https://patentlyo.com/patent/2018/06/eligibility-disclosed-embodiments.html.

[47] *See id.*

[48] *See* William B. Whitney, *Patentable Processes*, 19 HARV. L. REV. 30, 30 (1905).

[49] One author of the 1952 Patent Act commented:

> While patents are creatures of statute, the entire body of patent law is much fuller than the statute itself, including a vast amount of case material on subjects such as invention and infringement which are dealt with by the statute only in general terms. Consequently, a discussion of the statute alone cannot be a complete dissertation on patent law.

never explained the meaning of terms like "process" or "machine" under § 101, nor has it delegated authority to the Patent Office to interpret the patent statutes through regulation.[50] As a result, most substantive standards for patentability are judicially created.[51] Indeed, U.S. patent law has largely been "built upon judicial interpretation of elliptical statutory phrases, or is devoid of any statutory basis whatsoever."[52]

This section starts with nineteenth century judicial decisions that still guide the modern judicial approach to patent eligibility and functional claiming. Before the 1952 Patent Act, courts would invalidate claims altogether if the claims were either abstract or purely functional.[53] That changed with the 1952 Patent Act, through which Congress tried to soften the impact of functional claiming under § 112(f).[54] The idea was to allow functional claims, but limit their scope to the specific embodiments disclosed in the patent.[55] Courts, however, interpreted § 112(f) to cover only patents that invoked this statutory provision through specific claim language.[56] This approach allowed many patents, especially in the software field, to use functional claim language without triggering the narrowing effects of § 112(f).[57]

A. *Common Law Origins*

Starting in the mid-nineteenth century, courts began to distinguish between unpatentable "principles" and specific

P.J. Federico, *Commentary on the New Patent Act*, *reprinted in* 75 J. PAT. & TRADEMARK OFF. SOC'Y, 161, 162 (1993).

[50] *See* Tafas v. Doll, 559 F.3d 1345, 1352 (Fed. Cir. 2009), *vacated en banc on other grounds*, 328 F. App'x 658 (Fed. Cir. 2009).

[51] *See* Craig Allen Nard, *Legal Forms and the Common Law of Patents*, 90 B.U. L. REV. 51, 54 (2010).

[52] *Id.*

[53] *See* Lemley, *supra* note 26, at 914–15.

[54] *See id.* at 915–16.

[55] *See id.*

[56] *See* Lighting World, Inc. v. Birchwood Lighting, Inc., 382 F.3d 1354, 1358 (Fed. Cir. 2004), *overruled by* Williamson v. Citrix Online, LLC, 792 F.3d 1339, 1348–49 (Fed. Cir. 2015); Flo Healthcare Solutions, LLC v. Kappos, 697 F.3d 1367, 1374 (Fed. Cir. 2012).

[57] Lemley, *supra* note 26, at 923–24.

applications of such principles.[58] In *Le Roy v. Tatham*,[59] the Supreme Court explained that "[a] principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right."[60] By contrast, "[a] new property discovered in matter, *when practically applied*, in the construction of a useful article of commerce or manufacture, is patentable."[61] To illustrate this distinction, the Court explained that:

> Through the agency of machinery a new steam power may be said to have been generated. But no one can appropriate this power exclusively to himself, under the patent laws. The same may be said of electricity, and of any other power in nature, which is alike open to all . . . .[62]

Around the same time, courts developed a closely related doctrine to prohibit functional claiming. Early decisions often called this the "function of the machine" doctrine, and this doctrine held that machines could only be patented based on their components or design, not on the result or effect of a machine's operation.[63] The doctrine can be traced back to the Supreme Court's decision in *Corning v. Burden*,[64] where the court held an inventor could not patent "the function of a machine, or the effect produced by it on the material subjected to the action of the machine."[65] The Court reasoned the scope of functional claims exceeds the inventor's contribution to the technical field.[66] That is, functional claim language allows the inventor to "describe a machine which will perform a certain function, and then claim the function itself, and all other machines that may be invented to perform the same function."[67]

Historically, the distinction between functional claiming and subject matter eligibility has never been clear. Nineteenth century

[58] Le Roy v. Tatham, 55 U.S. 156, 175 (1852); Wyeth v. Stone, 30 F. Cas. 723, 727 (C.C.D. Mass. 1840).

[59] *Le Roy*, 55 U.S. 156.

[60] *Id.* at 175.

[61] *Id.* (emphasis added).

[62] *Id*.

[63] *See* Diamond v. Diehr, 450 U.S. 175, 197 (1981) (Stevens, J., dissenting).

[64] Corning v. Burden, 56 U.S. 252 (1853).

[65] *Id.* at 268.

[66] *See id.*

[67] *Id.* at 269.

cases that invalidate patents for reciting abstract principles are often based, at least in part, on functional claim language.[68] This is partly because the distinction between functional claiming and subject matter eligibility was hardly significant at that time. Before the 1952 Patent Act, courts would invalidate patents if they claimed either "the function of a machine" or ineligible subject matter.[69] For example, in *Wyeth v. Stone*, the court invalidated a claim that covered any machine for cutting ice.[70] Justice Story, riding circuit, reasoned that "[n]o man can have a right to cut ice by all means and methods."[71] Although a machine that cuts ice is neither an abstract principle nor a natural law, Justice Story nevertheless characterized the claim as "an art or principle in the abstract" because it did not relate to "any particular method or machinery" for cutting ice.[72]

Even in the renowned case of *O'Reilly v. Morse*,[73] it is not entirely clear how much weight the Supreme Court placed on functional claiming concerns.[74] There, the Supreme Court held Samuel Morse could not patent the use of electric current to generate characters or signs at a remote location, which is the principle behind the telegraph.[75] Morse's patent recited the following invention:

> I do not propose to limit myself to the specific machinery or parts of machinery described in the foregoing specification and claims; the essence of my invention being the use of the motive power of the electric or galvanic current, which I call electro-magnetism, however developed for marking or printing intelligible characters, signs, or letters, at any distances, being a new application of that power of which I claim to be the first inventor or discoverer.[76]

As his patent makes explicit, Morse tried to claim the use of electromagnetism to generate intelligible characters at a distance,

[68] *See, e.g.*, Wyeth v. Stone, 30 F. Cas. 723, 727 (C.C.D. Mass. 1840); O'Reilly v. Morse, 56 U.S. 62 (1853).

[69] *Wyeth*, 30 F. Cas. at 727; Lemley, *supra* note 26, at 914–15.

[70] *Wyeth*, 30 F. Cas. at 727.

[71] *Id.*

[72] *Id.*

[73] O'Reilly v. Morse, 56 U.S. 62 (1853).

[74] *See generally id.*

[75] *See id.* at 113.

[76] *Id.* at 112.

not just a specific machine for doing so.[77] The Supreme Court observed that the claim would give Morse the "exclusive right to every improvement where the motive power is the electric or galvanic current, and the result is the marking or printing intelligible characters, signs, or letters at a distance."[78] Morse's patent could be asserted against future inventors who "discover[ed] a mode of writing or printing at a distance by means of the electric or galvanic current, without using any part of the process or combination set forth in the plaintiff's specification."[79] As a result, the Court held that Morse's "claim is too broad, and not warranted by law."[80]

O'Reilly v. Morse remains a cornerstone of judicial interpretations of patentable subject matter.[81] It is cited in numerous Supreme Court decisions on patent eligibility, usually for the proposition that scientific principles cannot be patented.[82] The Federal Circuit has likewise described *O'Reilly v. Morse* as a patent eligibility case.[83] But is this characterization accurate? Samuel Morse tried to patent the use of electromagnetism in telegraphy. Electromagnetism does not fit within any of the categories of patent-eligible subject matter. Nor is electromagnetism "new," since it existed in nature long before Samuel Morse harnessed its power to transmit messages. But Morse's patent recites more than the bare principles of electromagnetism. Samuel Morse applied electromagnetism to generate characters at a remote location, which is a phenomenon that does not occur in nature and cannot exist

[77] *See id.*

[78] *Id.*

[79] *Id.* at 113.

[80] *Id.*

[81] *See, e.g.*, Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2354 (2014); Mayo Collaborative Servs. v. Prometheus Labs., Inc., 566 U.S. 66, 85 (2012); Parker v. Flook, 437 U.S. 584, 592 (1978); Gottschalk v. Benson, 409 U.S. 63, 68 (1972).

[82] *See, e.g.*, *Alice*, 134 S. Ct. at 2354; *Mayo*, 566 U.S. at 85; *Parker*, 437 U.S. at 592; *Gottschalk*, 409 U.S. at 68.

[83] *See, e.g.*, Finjan, Inc. v. Blue Coat Sys., Inc., 879 F.3d 1299, 1305 (Fed. Cir. 2018); Amdocs (Israel) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1309 (Fed. Cir. 2016); Intellectual Ventures I LLC v. Symantec Corp., 838 F.3d 1307, 1329 (Fed. Cir. 2016); McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1312 (Fed. Cir. 2016); Ariosa Diagnostics, Inc. v. Sequenom, Inc., 809 F.3d 1282, 1290 (Fed. Cir. 2015).

without human ingenuity.[84] Further, converting electrical impulses to legible characters is plainly a "process," at least in common parlance.[85]

Viewed this way, the rationale in *Morse* is more consistent with the judicial prohibition against functional claiming. Morse's patent was directed to the function of generating characters at a distance using electricity without regard to the specific *machine or process* for doing so. The Court explained that "[i]f this claim can be maintained, it matters not by what process or machinery the result is accomplished."[86] Future inventors might develop a device that is "less complicated[,] . . . less expensive in construction, and in its operation. But yet if it is covered by this patent the inventor could not use it, nor the public have the benefit of it without the permission of this patentee."[87] Indeed, the Supreme Court upheld different claims in Morse's patent that limited the claimed method to the process described in patent specifications.[88]

Aside from its tendency to blur into a subject matter eligibility analysis, the judicial bar on functional claims presented other conceptual difficulties as it evolved in the nineteenth century. First, there is tension between the prohibition on functional claiming and how courts interpret process patents. In *O'Reilly v. Morse*, the Supreme Court observed that Morse's patent improperly preempts future machines from performing the claimed process of generating

[84] *See Morse*, 56 U.S. 62.

[85] *See* Michael Risch, *America's First Patents*, 64 FLA. L. REV. 1279, 1320 (2012) (summarizing early American process patents); *see also* Cochrane v. Deener, 94 U.S. 780, 788 (1876) ("If [a process is] new and useful, it is just as patentable as is a piece of machinery. In the language of the patent law, it is an art.").

[86] *Morse*, 56 U.S. at 113.

[87] *Id.*

[88] *See, e.g.*, *id.* at 112 ("We perceive no well–founded objection to the description which is given of the whole invention and its separate parts, nor to his right to a patent for the first seven inventions set forth in the specification of his claims."). The Supreme Court's later decision in the Telephone Cases support this interpretation. Dolbear v. Am. Bell Tel. Co., 8 S. Ct. 778, 785 (1888) (upholding Alexander Graham Bell's patent on a process of transmitting speech using electricity because the claim was limited to the particular process discussed in the patent).

characters at a distance using electromagnetism.[89] But *all process patents* preempt others from using different machines to achieve the same result, including innovative machines that might perform the same process faster and cheaper.[90] In *Cochrane v. Deener*,[91] the Supreme Court found the defendants infringed a process patent for manufacturing flour, despite using a different machine.[92] The Court noted "a process may be patentable, irrespective of the particular form of the instrumentalities used" and "[i]f one of the steps of a process be that a certain substance is to be reduced to a powder, it may not be at all material what instrument or machinery is used to effect that object, whether a hammer, a pestle and mortar, or a mill."[93]

Second, the notion of what constitutes a functional claim can be elusive.[94] Even structural components are defined by their function to some extent.[95] Suppose that, in my hypothetical example of an airplane patent, the inventor claims "a flying machine with a fuselage, two wings, and a propeller." At first glance, nothing about this claim seems "functional," as it apparently describes the physical components of a propeller plane. But in reality, whether this claim is functional depends on how the patent defines the term "propeller." Conceivably, the patent might broadly define the term as "a device that propels," in which case the claim would still cover future innovations such as jet engines and other innovative means of propelling a plane. Even common definitions of "propeller" still include functional aspects. For instance, Merriam-Webster defines the term "propeller" as a device with rotating blades that "forms part

[89] *Morse*, 56 U.S. at 113.

[90] Cochrane v. Deener, 94 U.S. 780, 787 (1876).

[91] Cochrane v. Deener, 94 U.S. 780 (1876).

[92] *Id.* at 787–90.

[93] *Id.* at 787–88.

[94] Application of Fuetterer, 319 F.2d 259, 263 n.9 (C.C.P.A. 1963) ("One of the primary problems we have in coming to grips with the instant rejection is in what sense the word 'functional' is being used. Few words in patent law have acquired more diverse meanings than the word 'functional.'").

[95] B. L. Zangwill, *Comments on Means Claims and Expressions*, 34 J. PAT. OFF. SOC'Y 36, 36 (1952) ("[T]here is little, if anything, to guide us as to where 'structure' ends and 'function' begins, or even why one is always to be preferred to the other.").

of a helical surface and that is *used to propel a vehicle*."[96] Nevertheless, terms like "propeller" usually connote a minimum level of structure, even if such terms have functional aspects as well.[97] As described in Section III, however, such structural elements fall away entirely for software.[98]

By the early twentieth century, courts shifted towards a somewhat different rationale for invalidating functional claims. Rather than focusing on how functional claims tie up natural laws or abstract ideas, courts reasoned that functional claims do not define the scope of the invention with adequate clarity.[99] This reasoning is based on a long-standing patent law doctrine commonly known as *definiteness*.[100] As early as the Patent Act of 1790, patents were required to describe the invention and "distinguish the invention or discovery from other things before known and used."[101] Similarly, the Patent Act of 1870 required claims that "particularly point out and distinctly claim the part, improvement, or combination which he claims as his invention or discovery . . . ."[102]

In *Halliburton Oil Well Cementing Co. v. Walker*,[103] the Supreme Court relied on the definiteness requirement to hold that patents cannot claim an invention "in terms of what it will do rather than in terms of its own physical characteristics or its arrangement in the new combination apparatus."[104] By claiming the invention in terms of "what it will do," the Court found functional claiming undermines at least three policies rationales for the definiteness requirement:

> 1. That the Government may know what they have granted and what will become public property when the term of the monopoly expires.

[96] *Propeller*, MERRIAM–WEBSTER (emphasis added), https://www.merriam-webster.com/dictionary/propeller (last visited Aug. 8, 2018).

[97] *See* Lemley, *supra* note 26, at 960.

[98] *See infra* § III.B.

[99] *See* Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 9 (1946); Holland Furniture Co. v. Perkins Glue Co., 277 U.S. 245, 256–57 (1928).

[100] Nautilus, Inc. v. Biosig Instruments, Inc., 134 S. Ct. 2120, 2124–25 (2014).

[101] Act of Apr. 10, 1790, § 2, 1 Stat. 110.

[102] William Redin Woodward, *Definiteness and Particularity in Patent Claims*, 46 MICH. L. REV. 755, 760 (1948).

[103] *Halliburton*, 329 U.S. at 9.

[104] *Id.* at 9.

> 2. That licensed persons desiring to practice the invention may know, during the term, how to make, construct, and use the invention.
>
> 3. That other inventors may know what part of the field of invention is unoccupied.[105]

As a result, the claim in *Halliburton* was not invalidated because it was abstract, but because it failed to adequately describe the scope of the invention.[106] Stated otherwise, the Court found that functional claims fail to meet a notice requirement.[107] Although *Halliburton* was understood to have ended the practice of functional claiming, this prohibition on functional claiming would soon be legislatively abrogated by the 1952 Patent Act.[108]

B. *The 1952 Patent Act*

In 1926, Congress started to codify the laws of the United States into fifty titles according to subject matter, which ultimately produced the United States Code.[109] The 1952 Patent Act was part of this "comprehensive program of revising and enacting into law all of the titles of the United States Code."[110] Congress, however, went beyond codifying existing patent statutes and judicial decisions; it also revised patent law in several important respects.[111] Among other changes, the 1952 Patent Act divided patentable subject matter and novelty into two statutory sections: § 101 and § 102, respectively.[112] The law also created § 103, which states that only non-obvious inventions can be patented.[113]

[105] *Id.* at 10.

[106] *Id.* at 12.

[107] *Id.*

[108] Lemley, *supra* note 26, at 915.

[109] *Detailed Guide to the United States Code*, OFFICE OF THE LAW REVISION COUNSEL, http://uscode.house.gov/detailed_guide.xhtml (last visited July 31, 2018).

[110] H.R. REP. NO. 1923, at 1 (1952) (Conf. Rep.).

[111] Federico, *supra* note 49, at 164.

[112] *Id.* at 176.

[113] L. James Harris, *Some Aspects of the Underlying Legislative Intent of the Patent Act of 1952*, 23 GEO. WASH. L. REV. 658, 671–72 (1955); H.R. REP. NO. 1923, at 7 (1952) (Conf. Rep.) ("An invention which has been made, and which is new in the sense that the same thing has not been made before, may still not be patentable if the difference between the new thing and what was known before is not considered sufficiently great to warrant a patent.").

The 1952 Patent Act, however, did not substantively alter the statutory language regarding patentable subject matter. Section 101 provides that "[w]hoever 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 . . . ."[114] Compared to prior versions, the 1952 Patent Act replaced the term "art" with "process."[115] This change was not meant to be substantive because the term "art" had already "been interpreted by the courts as being practically synonymous with process or method."[116] The statute did, however, explicitly define "process" to include "a new use of a known process, machine, manufacture, composition of matter, or material."[117] The purpose for including this definition was to abrogate case law that cast doubt on whether new uses of existing compounds or devices are patentable.[118] By revising the definition of "process" to include new uses of existing processes or machines, Congress made clear that new uses of existing processes or machines would fall within the realm of patent-eligible subject matter.[119]

Congress also liberalized the use of functional claims. In *Halliburton*, the Supreme Court held patents cannot claim the invention "in terms of what it will do."[120] Although this decision was largely consistent with the Supreme Court's precedents,[121] it drove Congress to craft legislation that would allow functional claiming while also addressing the Court's concerns about the scope of such claims.[122] Under the 1952 Patent Act, paragraph six of § 112 states:

[114] 35 U.S.C. § 101 (2012).

[115] *Id.* at Reviser Notes.

[116] *Id.*

[117] 35 U.S.C § 100(b) (2012); CLS Bank Int'l v. Alice Corp., 717 F.3d 1269, 1295 (Fed. Cir. 2013) ("The 1952 Act shows that the 'primary significance' of adding Section 100(b) was to make clear that a method was not 'vulnerable to attack, on the ground of not being within the field of patentable subject matter . . . .'") (internal citations omitted).

[118] *CLS Bank*, 717 F.3d at 1294–95.

[119] Federico, *supra* note 49, at 175–77.

[120] Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1, 9 (1946).

[121] *See, e.g.*, Holland Furniture Co. v. Perkins Glue Co., 277 U.S. 245, 256–57 (1928) (listing Supreme Court precedents regarding patentable subject matter).

[122] Federico, *supra* note 49, at 186.

> 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.[123]

As the statute makes clear, a claim may cover a function without reciting a structure, material or act to support it. The caveat, however, is "such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof."[124] In other words, a patentee can use functional language, but the claim will be limited to whatever components or devices disclosed in the patent. Accordingly, an inventor who makes a propeller plane can still claim "a machine with a fuselage, wheels, and a means for flying." Under § 112, the claim is not invalid even though "means for flying" is functional. Courts will, however, limit the claim's scope to the propeller plane and its equivalents.[125] That way, the inventor cannot assert her patent against future inventors who create different "means for flying."

III. THE CHALLENGE OF SOFTWARE PATENTS

As Congress crafted 1952 Patent Act, a technological revolution was already afoot.[126] In 1946, scientists at the University of Pennsylvania designed the first general-purpose digital computer, called the ENIAC.[127] The same year Congress passed the 1952 Patent Act, scientists at the Los Alamos Scientific Laboratory

[123] 35 U.S.C. § 112, ¶ 6 (2011) [hereinafter § 112(f)]. The America Invents Act reorganized Section 112, and paragraph six is now codified under subsection (f). Leahy–Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011). The wording of this statutory subsection remains unchanged.

[124] 35 U.S.C. § 112(f) (2011).

[125] *See id.* (explaining that a functional claim limitation shall be "construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof").

[126] David Bender, *Computer Programs: Should They be Patentable*, 68 COLUM. L. REV. 241, 243–44 (1968).

[127] Frank da Cruz, *Programming the ENIAC*, COLUM. U. COMPUTING HIST. (Apr. 24, 2017), http://www.columbia.edu/cu/computinghistory/eniac.html.

created MANIAC I, a computer that could function on stored programs instead of hard-wired circuitry.[128]

Software and general-purpose computers present difficult issues for patent eligibility. Unlike mechanical or chemical processes, software does not cause a physical transformation such as the creation of a new machine or chemical compound. Instead, computer programs manipulate data and signals. And, unlike traditional machines, most computer functions are untethered to their specific hardware configurations.[129] General-purpose computers usually have a fairly standard set of components like a central processing unit and memory units, and most software programs will run on any standard hardware configuration.[130]

When drafting the 1952 Act, Congress likely could not have predicted the impact of digital computers, or how difficult it would be to fit them under the existing patent jurisprudence. Accordingly, the law did not include any provisions to address software patents. But, not long after the 1952 Patent Act became law, software began to fundamentally change the paradigm for how machines function. Since then, courts have looked for a workable approach to assess the patentability of computer programs. This search has shaped the modern judicial approach to both patent eligibility and functional claiming.

A. *Patent Eligibility of Software Patents*

By the mid-1960s, computer technology was becoming ubiquitous, and policymakers started to recognize the difficult issues it created for patent law. In 1965, President Johnson established a commission to address emerging issues raised by "complex and rapidly changing technology" and make recommendations for reforming the patent system.[131] The Commission proposed

[128] Diamond v. Diehr, 450 U.S. 175, 194 n.1 (1981) (Stevens, J., dissenting).

[129] *General Purpose Computer*, INST. FOR TELECOMM. SCIS., https://www.its.bldrdoc.gov/fs-1037/dir-017/_2452.htm (last visited Aug. 10, 2018).

[130] *Computer Hardware*, STAN. U., https://web.stanford.edu/class/cs101/hardware-1.html (last visited Aug. 4, 2018).

[131] Order Establishing the President's Commission on the Patent System, 30 Fed. Reg. 4661 (1965).

amending the Patent Act to categorically exclude software from patenting. Under the Committee's proposal:

> A series of instructions which control or condition the operation of a data processing machine, generally referred to as a "program," shall not be considered patentable regardless of whether the program is claimed as: (a) an article, (b) a process described in terms of the operations performed by a machine pursuant to a program, or (c) one or more machine configurations established by a program. [132]

The Commission noted "[u]ncertainty now exists as to whether the statute permits a valid patent to be granted on programs."[133] And it advocated against software patents in part because "the creation of programs has undergone substantial and satisfactory growth in the absence of patent protection and that copyright protection for programs is presently available."[134] The Commission's recommended fix would not only preclude claiming programs in the abstract, but would also exclude claims on programmable devices built for specific tasks.

Based on the Commission's recommendations, Congress introduced the Patent Reform Act of 1967, which would have amended the 1952 Act by adding § 106, titled "Computer programs not patentable."[135] This proposed amendment stated "[a] plan of action or set of operating instructions, in whatever form presented, to cause a controllable data processor or computer to perform selected operations shall not be patentable."[136] Echoing the recommendation of the Commission's Report, this statutory amendment would have eliminated patent protection for software entirely.

[132] STAFF OF S. COMM. ON THE JUDICIARY, 90TH CONG., REP. ON THE PRESIDENT'S COMM'N ON THE PATENT SYS. 20 (1966).

[133] *Id.* at 21.

[134] *Id.*

[135] Bender, *supra* note 126, at 241–42; Donald W. Banner, *The Recent Proposal to Change the United States Patent System*, 29 OHIO ST. L.J. 873, 876 (1968); *Patent Law Revision*, *Hearings on S.2, S. 1042, S. 1377 and S. 1691 Before the Subcomm. on Patents, Trademarks, and Copyrights of the S. Comm. on the Judiciary*, 90th Cong. 9 (1967) (statement of Sen. John L. McClellan, Chairman, Subcomm. on Patents, Trademarks, and Copyrights) [hereinafter *Patent Law Revision Hearings*].

[136] *See* Bender, *supra* note 126, at 241–42.

Not surprisingly, the nascent software industry vigorously opposed a categorical prohibition on patenting computer programs. Some argued it would be illogical to distinguish between traditional machines controlled by hardware and machines controlled by "a complex set of coded electrical impulses."[137] For example, the president of Applied Data Research, one of the largest software vendors at the time, testified that "a machine containing a programmed control system is the same in all features as that containing special purpose hardware controls."[138] The Chamber of Commerce argued "[f]urther study should be given to alternative solutions for determining what is and what is not patentable in the field of computer software, instead of immediately removing a vast new body of technology from patent consideration."[139] And the Commerce Department noted the difficulty of defining "computer program" and argued that a legislative exclusion for computer programs would be premature.[140] Those opposing the amendment ultimately prevailed, and, despite holding extensive hearings on the proposed amendments, Congress did not pass the 1967 Patent Reform Act.

Without a legislative solution, courts were left to grapple with computer programs on a case-by-case basis. In *Gottschalk v. Benson*, the Supreme Court held that a method of converting binary coded decimals to pure binary numerals was not patent-eligible.[141] Because the claims covered an algorithm in the abstract, the court observed "the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself."[142] The Court also stressed how "[t]he mathematical procedures can be carried out in existing computers long in use, no

[137] *Patent Law Revision Hearings*, *supra* note 135, at 751–53 (statement of Richard C. Jones, President of Applied Data Research Inc.).

[138] *See id.* at 751.

[139] *Id.* at 454 (statement of George F. Metcalf on behalf of the Chamber of Commerce of the United States).

[140] *Id.* at 724 (statement of Pedro R. Vazquez for the General Counsel of the U.S. Department of Commerce).

[141] *See* Gottschalk v. Benson, 409 U.S. 63, 64 (1972).

[142] *Id.* at 72.

new machinery being necessary. And, as noted, they can also be performed without a computer."[143]

Intuitively, the Court's holding that algorithms are abstract makes sense. After all, someone could use a pen and paper to perform decimal conversions or perhaps even make the calculations in their head. The more challenging question is whether specific computer implementations of software algorithms are patent-eligible. For example, could someone patent a method of using a computer to run an algorithm that pools mutual funds into an investment portfolio?[144] Because it runs on computer hardware, this patent might be less abstract than a method for converting binary signals to decimal. Arguably, however, the only "innovative" aspect of the claimed invention is the abstract idea of pooling mutual funds, not the computer that runs it.

Until recently, the Supreme Court's approach to specific implementations of software algorithms was unclear and inconsistent. In *Parker v. Flook*, the Court held that using an algorithm to automatically adjust variables in a chemical reaction was not patent-eligible.[145] In doing so, the court rejected the notion that "if a process application implements a principle in some specific fashion, it automatically falls within the patentable subject matter of § 101."[146] Yet the Supreme Court reached a different conclusion in *Diamond v. Diehr*, where the Court held that applying a mathematical equation to the process of curing rubber was patent-eligible.[147] Even though the physical steps in the claimed process were well-known, the Court characterized the claim as a "process for molding rubber products and not as an attempt to patent a mathematical formula."[148] *Flook* and *Diehr* seemingly reached

[143] *Id.* at 67.

[144] *See, e.g.*, State St. Bank & Tr. Co. v. Signature Fin. Grp., Inc., 149 F.3d 1368, 1373 (Fed. Cir. 1998) (upholding patent on computerized accounting system for managing mutual funds), *abrogated by* Bilski v. Kappos, 561 U.S. 593, 600 (2010) (invalidating patent for computer implemented method of hedging risk in commodity markets).

[145] Parker v. Flook, 437 U.S. 584, 594–95 (1978).

[146] *Id.* at 593.

[147] *See* Diamond v. Diehr, 450 U.S. 175, 186 (1981).

[148] *Id.* at 191.

opposite conclusions about whether implementing an algorithm in a specific application is patent-eligible, and the tension between the two cases has been extensively examined in the legal literature.[149]

Given this muddled Supreme Court guidance, lower courts took a fairly liberal approach to software patents. For its part, the Federal Circuit held an abstract method implemented on a generic computer is patent-eligible because it produces "a useful, concrete and tangible result."[150] Under this approach, the court found a computer-implemented method for pooling mutual funds was patentable because it transforms "data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price."[151] The Federal Circuit's approach helped fuel a sharp rise in software patenting that started in the 1990s and continued into the 2000s.[152] By 2011, the Government Accountability Office (GAO) estimated over half of all issued patents were related to software.[153] The same GAO study also found software patents accounted for more than half of all patent lawsuit filings.[154]

B. *Functional Claiming in Software Patents*

At the same time, the judicial approach to functional claiming under § 112 also promoted the growth of software patents. Section 112(f) applies to claims that express an element "as a means or step for performing a specified function without the recital of structure,

[149] *See, e.g.*, *Diehr*, 450 U.S. at 219 (Stevens, J., dissenting) (noting the inevitable confusion these decisions would create, and urging the Court to adopt an "unequivocal holding that no program–related invention is a patentable process under § 101 unless it makes a contribution to the art that is not dependent entirely on the utilization of a computer"); John M. Golden, *Flook Says One Thing, Diehr Says Another: A Need for Housecleaning in the Law of Patentable Subject Matter*, 82 GEO. WASH. L. REV. 1765 (2014).

[150] State St. Bank & Tr. Co. v. Signature Fin. Grp., Inc, 149 F.3d 1368, 1373 (Fed. Cir. 1998).

[151] *Id.*

[152] Hylton, *supra* note 34, at 1125–26.

[153] U.S. GOV'T ACCOUNTABILITY OFFICE, GAO-13-465, INTELLECTUAL PROPERTY: ASSESSING FACTORS THAT AFFECT PATENT INFRINGEMENT LITIGATION COULD HELP IMPROVE PATENT QUALITY 13 (2013).

[154] *Id.*; Hylton, *supra* note 34, at 1126.

material, or acts in support thereof."[155] Until recently, the Federal Circuit took this language quite literally. That is, § 112(f) presumably did not apply unless the patent recited the terms "means for" or "step for."[156] As the court explained, "[w]hen the claim drafter has not signaled his intent to invoke § 112 [(f)], by using the term 'means,' we are unwilling to apply that provision without a showing that the limitation essentially is devoid of anything that can be construed as structure."[157]

At first glance, software patents may not seem functional because they recite hardware components and software constructs. These limitations, however, often fail to constrain software patents beyond the claimed functions. A typical software patent might claim "a computer readable medium for performing a set of functions."[158] The problem is "computer readable medium" does not impose any meaningful constraint on claim scope.[159] Software runs on computers, so it is always stored on a computer readable medium. Likewise, software patents often recite limitations that look structural but are actually purely functional. For example, the patent in *Finjan v. Secure Computing Corporation* covered a "system comprising . . . an interface . . . , a comparator . . . , and a logical engine."[160] Terms like "interface" and "comparator" have no structural aspect whatsoever and are software constructs that are defined entirely by their function.[161]

Although claims like "server comprising an interface and a logical engine" might cover function, they do not recite "means for" or "step for." Without these talismanic words, courts often refused to find the claims invoked § 112(f).[162] As a result, software patents

[155] 35 U.S.C. § 112(f) (2011).

[156] Flo Healthcare Solutions, LLC v. Kappos, 697 F.3d 1367, 1374 (Fed. Cir. 2012); Lighting World, Inc. v. Birchwood Lighting, Inc., 382 F.3d 1354, 1358 (Fed. Cir. 2004), *overruled by* Williamson v. Citrix Online, LLC, 792 F.3d 1339, 1348–49 (Fed. Cir. 2015).

[157] *Flo Healthcare*, 697 F.3d at 1374.

[158] *See* Lemley, *supra* note 26, at 919–22.

[159] *Id.*

[160] Finjan, Inc. v. Secure Computing Corp., 626 F.3d 1197, 1204 (Fed. Cir. 2010).

[161] *Id.* at 1204.

[162] Lemley, *supra* note 26, at 923–24.

would often claim broad functions while avoiding the narrowing effect of § 112(f).[163] One study found every software patent asserted by non-practicing entities used some variety of functional language, and half of all software patents litigated by other companies used functional claiming as well.[164] Functional claiming also allows patentees to assert software patents against later developed technology. As a result, some have argued "software patents circumvent[] the limits the 1952 Act places on functional claiming."[165]

In 2015, partly in response to the proliferation of functional claiming in software, the Federal Circuit abandoned the "[strong] presumption that a limitation lacking the word 'means' is not subject to § 112 [(f)]."[166] In *Williamson v. Citrix Online*,[167] the court observed its presumption "is unwarranted, is uncertain in meaning and application, and has the inappropriate practical effect of placing a thumb on what should otherwise be a balanced analytical scale."[168] Instead, the court announced the standard for applying § 112 should be "whether the words of the claim are understood by persons of ordinary skill in the art to have a sufficiently definite meaning as the name for structure."[169]

After *Williamson*, courts seem more willing to find that software patents invoke means plus function under § 112. For instance, the Federal Circuit held "compliance mechanism" and "symbol generator" were both means-plus-function limitations, even though the claims never used the term "means."[170] *Williamson*, however, did not eliminate functional software claims altogether.[171] District courts

[163] *Id.*

[164] Chien & Karkhanis, *supra* note 32, at 40–41.

[165] Lemley, *supra* note 26, at 928.

[166] Williamson v. Citrix Online, LLC, 792 F.3d 1339, 1349 (Fed. Cir. 2015).

[167] Williamson v. Citrix Online, LLC, 792 F.3d 1339 (Fed. Cir. 2015).

[168] *Id.* at 1349.

[169] *Id.*

[170] *See* Advanced Ground Info. Sys., Inc. v. Life360, Inc., 830 F.3d 1341, 1347 (Fed. Cir. 2016); Media Rights Techs., Inc. v. Capital One Fin. Corp., 800 F.3d 1366, 1371 (Fed. Cir. 2015).

[171] lntellicheck Mobilisa, Inc. v. Wizz Sys., LLC, 2016 WL 1182150, at *19 (W.D. Wash. Mar. 28, 2016), Finjan, Inc., v. Proofpoint, Inc., 2015 WL 7770208, at *10–11 (N.D. Cal. Dec. 3, 2015).

have held that terms like "content processor for processing content" and "circuitry . . . for receiving the information" are not functional claims under § 112(f), even though the hardware in the claim imposes no meaningful limits on claim scope.[172]

Regardless, any change wrought by *Williamson* has been eclipsed by the Supreme Court's *Alice v. CLS Bank* decision.[173] The court in *Alice* ruled a generic computer implementation of a business method was a patent-ineligible abstract concept.[174] After *Alice*, courts will often invalidate functional software claims altogether before reaching the § 112(f) inquiry. In some ways, the expansion of the abstract idea exception under *Alice* has now subsumed the functional claiming inquiry.

IV. LIMITING SOFTWARE PATENTS UNDER SECTION 101

By the start of the twenty-first century policymakers were expressing greater skepticism about software patents.[175] Those opposed to software patents argued such patents disclosed nothing innovative, had unclear scope, and hampered innovation.[176] Critics also focused on businesses that licensed and litigated patents as their exclusive source of revenue.[177] In 2011, these so called non-practicing entities, or "patent trolls," sued over 5,000 firms at an estimated cost of over $29 billion.[178]

In response, Congress and courts tried to address these perceived abuses of the patent system. In 2011, Congress passed the America Invents Act (AIA), which created several administrative procedures

[172] *Finjan*, 2015 WL 7770208, at *10–11; *see lntellicheck*, 2016 WL 1182150, at *19.

[173] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014).

[174] *Id.* at 2358.

[175] *See* Colleen V. Chien, *Reforming Software Patents*, 50 HOUS. L. REV. 325, 352–64 (2012).

[176] Bessen, *supra* note 30; John F. Luman III & Christopher L. Dodson, *No Longer a Myth, the Emergence of the Patent Troll: Stifling Innovation, Increasing Litigation, and Extorting Billions*, 18 INTELL. PROP. & TECH. L. J., 1 no. 5 (May 2006).

[177] Luman & Dodson, *supra* note 176, at 1.

[178] Bessen, *supra* note 30.

that made it easier to challenge patents at the PTO.[179] Around the same time, both the Federal Circuit and the Supreme Court took an increasingly narrow view of patentable subject matter, particularly as it relates to software.[180] Of these decisions, the most important for software patents is *Alice*, in which the Supreme Court invalidated a method of mitigating financial risk on a computer.[181] In doing so, the court adopted a two-step framework for patent eligibility.[182] The first step asks whether the patent is directed to an abstract idea or other judicial exception to patentable subject matter.[183] If it does, then the court moves on to step two, where it determines whether the claim recites an "inventive concept."[184]

Shortly before the Supreme Court decided *Alice*, Professor Mark Lemley cautioned "the current trend is one that would invalidate a wide swath of software patent claims, particularly functional claims . . . [,] not because they are too broad, or indefinite, but because they are not the sort of thing that is patentable at all."[185] Four years after *Alice*, this warning seems prescient. Under *Alice* step one, courts frequently hold software functions are unpatentable abstract ideas.[186] The Federal Circuit even stated it was a "foundational patent law principle[] that a result, even an innovative result, is not itself patentable."[187] So in many cases, functional software claims are no longer narrowed under § 112(f), but are instead invalidated under § 101.

There is an ongoing debate about whether this approach to software patents helps or hinders innovation. However, the growing

[179] *See* Leahy–Smith America Invents Act, Pub. L. No. 112–29, 125 Stat. 284 (2011).

[180] *See* Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2357 (2014); *see also* Mayo v. Prometheus, 566 U.S. 66, 72 (2012); Bilski v. Kappos, 561 U.S. 593, 605 (2010); Bancorp Servs., L.L.C. v. Sun Life Assur. Co. of Canada, 687 F.3d 1266, 1281 (Fed. Cir. 2012); CyberSource Corp. v. Retail Decisions, Inc., 654 F.3d 1366, 1370 (Fed. Cir. 2011).

[181] *Alice*, 134 S. Ct. at 2357.

[182] *Id.* at 2355.

[183] *Id.*

[184] *Id.* at 2357.

[185] Lemley, *supra* note 26, at 962.

[186] Gugliuzza & Lemley, *supra* note 39, at 790.

[187] Finjan, Inc. v. Blue Coat Sys., Inc., 879 F.3d 1299, 1305 (Fed. Cir. 2018).

consensus is the two-step process for evaluating patent eligibility fails to provide sufficient clarity for lower courts, the PTO, and practitioners.[188] This section does not attempt to resolve the decades-long debate about software patents, nor does it advocate for a new interpretation of § 101. It will, however, clarify the court's approach to functional software claims, which is critical to understanding the rationale underlying post-*Alice* judicial decisions. In particular, this section explains how courts cannot agree on whether software functions are inherently abstract. Some decisions hold if software claims use purely functional language, then they fail § 101 regardless of what the specification discloses.[189] By contrast, other opinions do not automatically condemn functional software claims.[190] Instead, they look to the patent specification to determine if the claimed software functions rely on a patent-eligible technological solution.[191]

[188] Interval Licensing LLC v. AOL, Inc., 896 F.3d 1335, at 1350 (Fed. Cir. 2018) (Plager, J., dissenting) ("That the phrase 'abstract ideas' is a definitional morass can be seen in one simple fact—a search for a definition of 'abstract ideas' . . . reveals that there is no single, succinct, usable definition anywhere available."); Berkheimer v. HP Inc., 890 F.3d 1369, 1374 (Fed. Cir. 2018) (Lourie, J., concurring) (per curiam) ("Individual cases, whether heard by this court or the Supreme Court, are imperfect vehicles for enunciating broad principles because they are limited to the facts presented. Section 101 issues certainly require attention beyond the power of this court.").

[189] *See, e.g.*, Apple, Inc. v. Ameranth, Inc., 842 F.3d 1229, 1241 (Fed. Cir. 2016); Affinity Labs of Texas, LLC v. Amazon.com Inc., 838 F.3d 1266, 1269 (Fed. Cir. 2016); Elec. Power Grp., LLC v. Alstom S.A., 830 F.3d 1350, 1356 (Fed. Cir. 2016); McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1314 (Fed. Cir. 2016) ("We therefore look to whether the claims in these patents focus on a specific means or method that improves the relevant technology or are instead directed to a result or effect that itself is the abstract idea and merely invoke generic processes and machinery.").

[190] *See, e.g.*, *Berkheimer* v. HP Inc., 881 F.3d 1360, 1371 (Fed. Cir. 2018); Visual Memory LLC v. NVIDIA Corp., 867 F.3d 1253, 1256 (Fed. Cir. 2017); Amdocs (Israel) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1291 (Fed. Cir. 2016).

[191] *Berkheimer*, 881 F.3d at 1371; *Visual Memory*, 867 F.3d at 1256; *Amdocs*, 841 F.3d at 1291.

A. *The Alice/Mayo Two-Step Framework*

The Supreme Court's opinion in *Alice* is commonly understood as a turning point for software patentability. While that may be true, *Alice* also represents the culmination of decades-long concerns regarding the proliferation of software patents, which were increasingly viewed as hampering innovation and exacting a toll on business. For its part, Congress passed the America Invents Act ("AIA") in 2011, which created new administrative procedures to make challenging overbroad patents easier.[192]

The AIA, however, did not change any substantive requirements for patentability.[193] Even without a legislative amendment to § 101, the Supreme Court issued a series of decisions that tightened patent eligibility standards.[194] In *Bilski v. Kappos*, the Supreme Court rejected the notion that a claimed process was patent-eligible so long as it was "tied to a particular machine or apparatus" or "transforms a particular article into a different state or thing."[195] The Court noted that while this "may well provide a sufficient basis for evaluating processes similar to those in the Industrial Age . . . , there are reasons to doubt whether the test should be the sole criterion for determining the patentability of inventions in the Information Age."[196]

The trend towards stricter patent eligibility standards continued in *Mayo v. Prometheus*, where the Supreme Court invalidated claims directed to a method for determining the proper dosage of a drug by measuring a specific biomarker.[197] *Mayo* is also the first case where the Court expressly adopted a two-step framework for determining patent eligibility. At step one, courts ask whether the claim as a whole is directed to a patent-ineligible concept.[198] If the claim is directed to a patent-ineligible concept, then the court

[192] Leahy–Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011).

[193] 35 U.S.C. § 101 (2012).

[194] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2357 (2014); Mayo v. Prometheus, 566 U.S. 66, 72 (2012); Bilski v. Kappos, 561 U.S. 593, 600 (2010).

[195] *Bilski*, 561 U.S. at 600. This was commonly known as the machine–or–transformation test. *Id.*

[196] *Id.* at 605.

[197] *Mayo*, 566 U.S. at 72–73.

[198] *Id.* at 77.

proceeds to step two, where it searches for "an 'inventive concept,' sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the natural law itself."[199]

Because *Mayo* involved a method of detecting a biomarker and adjusting drug dosage, its impact on computer and software patents was not immediately clear. In *Alice*, however, the Supreme Court confirmed *Mayo*'s two step approach applied to computer and software patents as well.[200] There, the Court held that implementing a method of mitigating financial risk on a generic computer is not patent-eligible.[201] Applying the two-step inquiry first announced in *Mayo*, the Court found the claimed financial method was an abstract idea at step one.[202] "[T]he claims . . . are drawn to the concept of intermediated settlement, i.e., the use of a third party to mitigate settlement risk," observed the Court.[203] And "the concept of intermediated settlement is 'a fundamental economic practice long prevalent in our system of commerce.'"[204]

At step two, the court held that applying an abstract idea on a computer was not sufficient to make that idea patent-eligible.[205] While recognizing that claims directed to a machine or computer-readable medium are "formally addressed to patent-eligible subject matter," the Court nevertheless held that "the mere recitation of a generic computer cannot transform a patent-ineligible abstract idea into a patent-eligible invention."[206] "Stating an abstract idea while adding the words 'apply it with a computer' simply combines those two steps, with the same deficient result."[207]

Notably, the analysis in *Alice* echoes earlier decisions on functional claiming. For instance, although the claims recited hardware such as "data processing system" and "communications

[199] *Id.*
[200] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2357 (2014).
[201] *Id.*
[202] *Id.*
[203] *Id.* at 2356.
[204] *Id.* (quoting Bilski v. Kappos, 561 U.S. 593, 611 (2010)).
[205] *Id.* at 2357.
[206] *Id.* at 2358.
[207] *Id.*

controller," the Court noted these components are "purely functional and generic." [208] The Court explained:

> Nearly every computer will include a "communications controller" and "data storage unit" capable of performing the basic calculation, storage, and transmission functions required by the method claims. As a result, none of the hardware recited by the system claims "offers a meaningful limitation beyond generally linking 'the use of the [method] to a particular technological environment,' that is, implementation via computers."[209]

After *Alice*, lower courts have generally held that claims reciting an abstract idea or algorithm implemented on a generic computer are ineligible for patenting.[210] One of the first post-*Alice* decisions from the Federal Circuit was *buySAFE, Inc. v. Google, Inc.*, in which the court invalidated a software patent directed to a "machine-readable media encoded to perform steps for guaranteeing a party's performance of its online transaction."[211] The court found "[t]he claims' invocation of computers adds no inventive concept" and "[t]he computer functionality is generic."[212] And the court emphasized that narrowing the claim to use in a computer is at best an "'attempt[] to limit the use of the abstract guarantee idea 'to a particular technological environment,' which has long been held insufficient to save a claim in this context."[213]

Some decisions go beyond finding that abstract ideas implemented on generic computers are ineligible, and they will even invalidate claims that recite specific devices or components under § 101.[214] For example, the patent in *In re TLI* related "a method and system for taking, transmitting, and organizing digital images."[215] Although some claims required "tangible components such as 'a telephone unit' and a 'server,'" the Federal Circuit determined "the

[208] *Id.* at 2360.
[209] *Id.* (alteration in original) (citations omitted).
[210] *See, e.g.*, OIP Techs., Inc. v. Amazon.com, Inc., 788 F.3d 1359, 1362 (Fed. Cir. 2015); Versata Dev. Grp., Inc. v. SAP Am., Inc., 793 F.3d 1306, 1333 (Fed. Cir. 2015); buySAFE, Inc. v. Google, Inc., 765 F.3d 1350, 1351 (Fed. Cir. 2014).
[211] *buySAFE*, 765 F.3d at 1351.
[212] *Id.*
[213] *Id.* at 1350.
[214] In re TLI Commc'ns LLC Patent Litig., 823 F.3d 607, 607 (Fed. Cir. 2016).
[215] *Id.*

recited physical components merely provide a generic environment in which to carry out the abstract idea of classifying and storing digital images in an organized manner."[216]

B. *Software Patents After Alice*

The overall impact of *Alice* on computer and software patents is significant. One study found the Federal Circuit invalidated patents in 90% of patent eligibility cases involving information technology.[217] For PTO reviews of covered business patents, which often involve software implementation of business methods, over 95% of all § 101 decisions resulted in patent invalidity.[218] In a concurring opinion, Judge Mayer urged his colleagues on the Federal Circuit "to acknowledge that *Alice* sounded the death knell for software patents."[219] He argued software "is inherently abstract because it is merely 'an idea without physical embodiment[.]' Given that an 'idea' is not patentable, and a generic computer is 'beside the point' in the eligibility analysis, all software implemented on a standard computer should be deemed categorically outside the bounds of § 101."[220]

Does Judge Mayer accurately claim that post-*Alice*, all software is ineligible under § 101? At least for now, the Federal Circuit has carved out a space for software patents that claim improvements in computer functionality. That is, the court distinguishes between improvements in computer technology itself, which remain patentable, and new functions of a conventional computer, which are ineligible.[221]

[216] *Id.* at 611.

[217] Gugliuzza & Lemley, *supra* note 39, at 767.

[218] *#Alicestorm: April Update and the Impact of TC Heartland on Patent Eligibility*, BILSKI BLOG, (June 1, 2017), https://www.bilskiblog.com/2017/06/alicestorm-april-update-and-the-impact-of-tc-heartland/#more-3237.

[219] Intellectual Ventures I LLC v. Symantec Corp., 838 F.3d 1307, 1325 (Fed. Cir. 2016) (Mayer, J., concurring).

[220] *Id.*

[221] Enfish, LLC v. Microsoft Corp., 822 F.3d 1327, 1335 (Fed. Cir. 2016) ("The Supreme Court has suggested that claims 'purport[ing] to improve the functioning of the computer itself,' or 'improv[ing] an existing technological process' might not succumb to the abstract idea exception."); *see also* Finjan, Inc. v. Blue Coat

To illustrate, consider the patent in *Enfish v. Microsoft*, which claimed a new logical model for a computer database.[222] The Federal Circuit reversed the district court's decision to invalidate the patent under § 101.[223] In doing so, the court stressed *Alice* did not "broadly hold that all improvements in computer-related technology are inherently abstract," and "some improvements in computer-related technology when appropriately claimed are undoubtedly not abstract, such as a chip architecture, an LED display, and the like."[224] The Court further noted "[s]oftware can make non-abstract improvements to computer technology just as hardware improvements can."[225] Thus, the relevant question is "whether the claims are directed to an improvement to computer functionality versus being directed to an abstract idea."[226] Turning to the patent at issue, the court held the claims were "specifically directed to a *self-referential* table for a computer database" that offered "increased flexibility, faster search times, and smaller memory requirements" over conventional computer databases.[227]

Nevertheless, it can be difficult to distinguish between "improvements in computer technology" and "new functions of generic computers." Under the Federal Circuit's standard, if a programmer writes an algorithm that makes a smartphone run faster or use less memory, then she has improved the computer's functionality.[228] By contrast, if the programmer develops an application that allows a smartphone to perform new tasks, she is

Sys., Inc., 879 F.3d 1299, 1304 (Fed. Cir. 2018); Thales Visionix Inc. v. United States, 850 F.3d 1343, 1348 (Fed. Cir. 2017).

[222] *Enfish*, 822 F.3d at 1335; *Finjan*, 879 F.3d at 1304; *Thales*, 850 F.3d at 1348.

[223] *Enfish*, 822 F.3d at 1339.

[224] *Id.* at 1335.

[225] *Id.*

[226] *Id.*

[227] *Id.* at 1337.

[228] *Id.* ("[O]ur conclusion that the claims are directed to an improvement of an existing technology is bolstered by the specification's teachings that the claimed invention achieves other benefits over conventional databases, such as increased flexibility, faster search times, and smaller memory requirements.").

likely implementing an abstract idea on generic components.[229] Arguably, however, a smartphone that functions as a payment device and a navigation tool is also an "improvement" over a device that only browses the Internet, even if the underlying hardware remains unchanged. Consider two cases:

1. Many commuters might prefer to pay their subway fare with a bank or credit card. The problem, however, is verifying bankcard transactions takes time, which inevitably creates delays in busy subway terminals.[230] In *Smart Systems*, the patentee solved this problem by storing a list of preapproved bank cards at each terminal.[231] That way, individual terminals can quickly determine whether a bank card is associated with a preapproved transit account by referencing the locally stored list.[232]

2. Network service providers need to monitor and account for the online activity of devices on their network.[233] But receiving and processing every device's network activity at a central server requires a lot of computational power and memory.[234] In *Amdocs*, the patentee claimed a system that records the activity of each networked device at or near the device's location.[235] This "reduces the storage and computational resource requirements" for the central server.[236]

The Federal Circuit reached different results in these two cases. In *Smart Systems*, the court invalidated a patent for verifying payments at a subway terminal.[237] In doing so, the court characterized the invention as "directed to the collection, storage,

[229] Apple, Inc. v. Ameranth, Inc., 842 F.3d 1229, 1241 (Fed. Cir. 2016) (affirming invalidity on patents covering a computer system that generates menus with specific features).

[230] Smart Sys. Innovations, LLC v. Chi. Transit Auth., 873 F.3d 1364, 1368 (Fed. Cir. 2017).

[231] *Id.* at 1372, 1379.

[232] *Id.*

[233] Amdocs (Israel) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1291 (Fed. Cir. 2016).

[234] *Id.*

[235] *Id.*

[236] *Id.* at 1308 (Reyna, J., dissenting).

[237] *Smart Sys. Innovations*, 873 F.3d at 1368.

and recognition of data."[238] Accordingly, the focus of the claims was not "'on the specific asserted improvement in computer capabilities.'"[239] Instead, the "computers are invoked merely as a tool" to carry out the claimed function.[240] In *Amdocs,* however, the court upheld a patent on a system of tracking network activity.[241] There, the court reasoned the claim aimed to solve the technological problem of "massive record flows" by applying the technological solution of "enhancing data in a distributed fashion."[242]

Are *Amdocs* and *Smart Systems* consistent with one another? Arguably, the patent in *Smart Systems* describes improvements in computer functionality. By storing a preapproved list of bank cards, the patent avoids network latency by verifying bank and credit card payments at transit terminals.[243] The court, however, characterized the claim as "the collection of financial data from third parties, the storing of that financial data, linking proffered credit cards to the financial data, and allowing access to a transit system based on the financial data."[244] But, as the dissent points out, this characterization arguably misses the "heart of the invention," which is overcoming the network latency that hinders the "use of conventional bankcards to access mass transit."[245]

Perhaps we can distinguish *Amdocs* from *Smart Systems* because monitoring network traffic seems more "technical" than storing a list of preapproved bank and credit cards. A human being can check bank-card numbers against a preapproved list at a subway terminal, however tedious or impracticable the task may be. However, no human can monitor network traffic without a computer. The problem with this reasoning is *Amdocs* did not involve a patent that actually claimed a technical solution for monitoring network

[238] *Id.* at 1372.

[239] *Id.* (quoting Enfish, LLC v. Microsoft Corp., 822 F.3d 1327, 1335 (Fed. Cir. 2016)).

[240] *Id.*

[241] *Amdocs*, 841 F.3d at 1291.

[242] *Id.* at 1300.

[243] *Smart Sys. Innovations*, 873 F.3d at 1380 (Linn, J., dissenting).

[244] *Id.* at 1372.

[245] *Id.* at 1382 (Linn, J., dissenting).

traffic.[246] One exemplary claim, for example, recites "a computer program product embodied on a computer readable storage medium for processing network accounting information" through distributed network architecture.[247] As the dissent observed, nothing in the claim recites how distributed network architecture would process network account information.[248]

Other than figuring out whether a patent is directed to an abstract concept at *Alice* step one, there are also practical challenges for determining whether software patents recite an inventive concept at step two. As a general matter, patent eligibility is a question of law.[249] It is also a threshold issue in many cases.[250] Accordingly, courts often decide patent eligibility at the pleadings stage before hearing any expert testimony or resolving claim construction.[251] Treating patent eligibility as a pure legal question, however, creates problems at *Alice* step two, where the court searches for an "inventive concept" in the claim.[252] The Supreme Court has stressed that an inventive concept must go beyond what is "well-understood, routine, conventional activity, previously engaged in by those in the field."[253] But how does a court determine what is well-understood,

[246] *Amdocs*, 841 F.3d at 1313 (Reyna, J., dissenting).

[247] *See id*. at 1299. Claim 1 recites in full:

> A computer program product embodied on a computer readable storage medium for processing network accounting information comprising:
> computer code for receiving from a first source a first network accounting record;
> computer code for correlating the first network accounting record with accounting information available from a second source; and
> computer code for using the accounting information with which the first network accounting record is correlated to enhance the first network accounting record.

Id.

[248] *Id.* at 1313–14.

[249] Bascom Glob. Internet Servs., Inc. v. AT&T Mobility LLC, 827 F.3d 1341, 1347 (Fed. Cir. 2016).

[250] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014); Mayo Collaborative Servs. v. Prometheus Labs., Inc., 566 U.S. 66, 75 (2012); Bilski v. Kappos, 561 U.S. 593, 602 (2010).

[251] Paul R. Gugliuzza, *Quick Decisions in Patent Cases*, 106 GEO. L. J. 619, 652 (2018).

[252] *Alice*, 134 S. Ct. at 2355.

[253] *Mayo*, 566 U.S. at 82.

routine and conventional? That inquiry necessarily depends on the state of the technology at the time of invention and requires courts to make factual determinations about the type of activity "engaged in by those in the field."[254]

Recognizing this problem, the Federal Circuit recently held *Alice* step two can include subsidiary factual issues.[255] In *Berkheimer v. HP Inc.*, the court reversed the district court's grant of summary judgment of invalidity under § 101.[256] The patent was directed to a process of storing digital files "without substantial redundancy."[257] The court held "[w]hether something is well-understood, routine, and conventional to a skilled artisan at the time of the patent is a factual determination."[258] And because the patent contends the claimed function "improves system operating efficiency and reduces storage costs," the court found "there is at least a genuine issue of material fact" as to whether the patents are directed to an improvement in computer functionality.[259] Similarly, in *Aatrix Software, Inc. v. Green Shades Software, Inc.*, the court vacated the district court's finding of invalidity because the amended complaint alleged a claimed "data file" improved the operability of the software.[260] Based on the amended complaint, the court held it was not proper to dismiss the claim on the pleadings.[261]

Treating patent eligibility as a factual issue, however, creates complications as well. Once courts delve into what technology was "well-understood" at the time of invention, the analysis starts to look like an obviousness determination under § 103. And resolving whether a claim is obvious usually requires a fact-intensive inquiry into the differences between the claimed invention and the prior

[254] *Id.*

[255] Aatrix Software, Inc. v. Green Shades Software, Inc., 882 F.3d 1121, 1126 (Fed. Cir. 2018); Berkheimer v. HP Inc., 881 F.3d 1360, 1371 (Fed. Cir. 2018).

[256] *Berkheimer*, 881 F.3d at 1369.

[257] *Id.* at 1370.

[258] *Id.* at 1369.

[259] *Id.* at 1370.

[260] *Aatrix*, 882 F.3d at 1126.

[261] *Id.* at 1126–27.

art.[262] Indeed, if courts were required to conduct an obviousness-type analysis under § 101, then it would seem incongruent to also treat patent eligibility as a threshold issue capable of resolution on the pleadings.

Judge Lourie recently identified some of the problems inherent in *Alice* step two and even questioned whether there should "be a step two in an abstract idea analysis at all."[263] In a concurring opinion to the denial of *en banc* rehearing in *Berkheimer*, Judge Lourie observed the prohibition on "computer functions [that] are 'well-understood, routine, conventional activit[ies]' previously known to the industry . . . is essentially a §§ 102 and 103 inquiry."[264] He further noted that, regardless of whether step two is treated as a question of fact or law, the decision will ultimately "not work us out of the current Section 101 dilemma," and the inquiry "digs the hole deeper by further complicating the Section 101 analysis."[265]

As Judge Lourie's opinion points out, the § 101 inquiry has become complex and unwieldy. The issue can be especially hard to resolve for the PTO, where patent examiners must decide whether a claimed invention was "routine and conventional" without access to discovery or expert testimony. To help guide this endeavor, the Deputy Commissioner for Patent Examination Policy issued a memo in 2018 that instructed examiners to rely on four sources to determine whether an idea is routine.[266] They are: (1) the express statement of the patent applicant; (2) court decisions; (3) a publication like a book or manual; or (4) official notice based on personal knowledge.[267] The last source, official notice, should only

[262] Graham v. John Deere Co. of Kansas City, 383 U.S. 1, 17 (1966); John M. Golden, *Patentable Subject Matter and Institutional Choice*, 89 TEX. L. REV. 1041, 1056–57 (2011).

[263] Berkheimer v. HP Inc., 890 F.3d 1369, 1375 (Fed. Cir. 2018) (Lourie, J., concurring).

[264] *Id.* at 1376 (alteration in original) (citations omitted).

[265] *Id.*

[266] U.S. PATENT AND TRADEMARK OFFICE, CHANGES IN EXAMINATION PROCEDURE PERTAINING TO SUBJECT MATTER ELIGIBILITY, RECENT SUBJECT MATTER ELIGIBILITY DECISION (BERKHEIMER V. HP, INC.) 3–4 (2018), https://www.uspto.gov/sites/default/files/documents/memo-berkheimer-20180419.PDF (last visited Oct. 29, 2018).

[267] *Id.*

be used where the examiner is certain the claimed method is "widely prevalent or in common use."[268] Practically speaking, however, figuring out if something is "widely prevalent or in common use" sounds like another way of asking whether the idea would have been especially obvious to those in the field.

C. *Addressing Functional Software Claims through Section 101*

Despite Judge Mayer's insistence that *Alice* sounded the death knell for software patents, at least some software patents remain valid.[269] But, as the previous section explains, figuring out which software patents are eligible requires wading into a morass of seemingly conflicting judicial decisions. And from a practical perspective, the *Alice* two-step test is hard to administer because it can encompass factual inquiries even though it is ostensibly a threshold issue.

This section looks at the *Alice* inquiry for software patents at a different angle. Instead of resolving whether claims recite "improvements on computer technology" or implement an abstract idea on generic computers, it focuses on how judicial concerns about functional software claims influence post-*Alice* decisions under § 101. Although viewing the issue from this perspective does not resolve every conceptual difficulty regarding software patents, it should at least clarify the current judicial approach to patent eligibility for software.

To start, there is general consensus that functional claim language is at least relevant to the § 101 inquiry post-*Alice*. In *Electric Power Group*, the Federal Circuit observed that "essentially result-focused, functional character of claim language has been a frequent feature of claims held ineligible under § 101, especially in the area of using generic computer and network technology to carry out economic transactions."[270] Similarly, the court in *Finjan v. Blue*

[268] *Id.* at 3.

[269] *See, e.g.*, Enfish, LLC v. Microsoft Corp., 822 F.3d 1327, 1335 (Fed. Cir. 2016) ("Nor do we think that claims directed to software, as opposed to hardware, are inherently abstract and therefore only properly analyzed at the second step of the Alice analysis.").

[270] Elec. Power Grp., LLC v. Alstom S.A., 830 F.3d 1350, 1356 (Fed. Cir. 2016).

Coat Systems announced that a result "is not itself patentable."[271] Even in *Amdocs*, which upheld the validity of the challenged patents, the majority acknowledged that functional claiming could be "a helpful way of double-checking" whether an invention is directed to an abstract idea or "an inventive concept in application."[272]

But beyond recognizing its relevance, courts cannot agree on a specific approach to functional software claims under § 101. Some decisions treat software functions as inherently abstract.[273] Accordingly, if the claims use purely functional language, then the patent likely fails § 101 regardless of what the specification discloses.[274] This approach "would save the patent's eligibility under § 101 only if the claim at issue itself explicitly states the necessary 'means'" to perform the claimed function.[275] By contrast, other decisions do not foreclose functional claiming altogether under § 101. Instead, they look to the patent specification to determine if the claimed function relies on a patent-eligible technological solution.[276] Under this latter approach, purely functional software claims can still survive § 101 scrutiny so long as the specification discloses technological improvements to perform the claimed function—even if those improvements are not expressly claimed.

To illustrate this distinction, I return to my hypothetical example of an inventor who creates a propeller plane, but broadly claims "a

[271] Finjan, Inc. v. Blue Coat Sys., Inc., 879 F.3d 1299, 1305 (Fed. Cir. 2018).

[272] Amdocs (Israel) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1295 (Fed. Cir. 2016); *see also* McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1314 (Fed. Cir. 2016) ("We therefore look to whether the claims in these patents focus on a specific means or method that improves the relevant technology or are instead directed to a result or effect that itself is the abstract idea and merely invoke generic processes and machinery.").

[273] Apple, Inc. v. Ameranth, Inc., 842 F.3d 1229, 1241 (Fed. Cir. 2016); Affinity Labs of Tex., LLC v. Amazon.com Inc., 838 F.3d 1266, 1269 (Fed. Cir. 2016); *Elec. Power Grp.*, 830 F.3d at 1356.

[274] *McRO*, 837 F.3d at 1314 ("A patent may issue 'for the means or method of producing a certain result, or effect, and not for the result or effect produced.'") (quoting Diamond v. Diehr, 450 U.S. 175, 182 n.7 (1981)).

[275] *Amdocs*, 841 F.3d at 1295.

[276] Berkheimer v. HP Inc., 881 F.3d 1360, 1371 (Fed. Cir. 2018); Visual Memory LLC v. NVIDIA Corp., 867 F.3d 1253, 1256 (Fed. Cir. 2017); *Amdocs*, 841 F.3d at 1295.

machine that applies Bernoulli's principle to fly." Under the first approach, we would treat the function of "applying Bernoulli's principle to fly" as an abstract idea. As a result, the claim is ineligible because it recites an abstract function, but is silent on how the invention applies Bernoulli's principle, what components are used, or how those components are assembled. By contrast, the second approach looks beyond the claims to the specification, which discloses a design for a propeller plane. Thus, the patent provides a non-abstract, mechanical device that applies Bernoulli's principle to fly. As a result, the claim is not abstract under the second approach because it captures the inventor's specific technological improvement.[277]

For its part, the Federal Circuit has yet to settle on one approach over the other. For example, *Apple v. Ameranth* reflects the first approach, in which software functions are considered inherently abstract.[278] There, the court invalidated a patent that disclosed a system for generating and transmitting menus that could be used in restaurants.[279] In doing so, the court observed "[t]he patents claim systems including menus with particular features. They do not claim a particular way of programming or designing the software to create menus that have these features, but instead merely claim the resulting systems. Essentially, the claims are directed to certain functionality . . . ."[280] Further, the court dismissed the specification's disclosure of programming details—the means for accomplishing the claimed function—as "immaterial because these details are not recited in the actual claims."[281] Similarly, in *Affinity Labs v. Amazon*, the Court invalidated claims that "describe a desired function or outcome, without providing any limiting detail that confines the claim to a particular solution to an identified problem. The purely

[277] *See, e.g.*, *Amdocs*, 841 F.3d at 1303 ("The collection, filtering, aggregating, and completing steps all depend upon the invention's unique distributed architecture—the same architecture outlined in our earlier analysis of the '065 patent. An understanding of how this is accomplished is only possible through an examination of the claims in light of the written description.").

[278] *Ameranth*, 842 F.3d at 1241.

[279] *Id.*

[280] *Id.*

[281] *Id.* at 1242.

functional nature of the claim confirms that it is directed to an abstract idea, not to a concrete embodiment of that idea."[282]

By contrast, other Federal Circuit decisions look beyond the claim language to determine whether the patent is directed to a non-abstract technological improvement.[283] For example, in *Visual Memory LLC v. NVIDIA Corp.*, the court upheld claims directed to a computer memory system with "programmable operational characteristics that can be tailored for use with multiple different processors . . . ."[284] The claims, however, said nothing about the "programmable operational characteristic," other than stating that it "determines a type of data stored by said cache."[285] Even though the claim described the computer memory system by its function, the court nevertheless found it patent-eligible in part because the patent disclosed code to perform the claimed function.[286]

Similarly, the court in *Amdocs* upheld the challenged patent because the claimed functions "all depend upon the invention's unconventional distributed architecture," which can only be understood "through an examination of the claims *in light of the written description*."[287] There, the claims recited functions like "collecting network communications usage information" and "storing the plurality of data records in a database."[288] But instead of focusing only on the functional nature of the claims, the court looked beyond the claim language and relied on portions of the specification that disclosed a "distributed architecture" to perform these functions.[289] *Amdocs* also expressly rejected the notion that a function, or a "desired goal," is always an abstract idea.[290] The court refused to "focus[] on the difference between 'means' and 'ends,'"

[282] Affinity Labs of Tex., LLC v. Amazon.com Inc., 838 F.3d 1266, 1269 (Fed. Cir. 2016).

[283] Visual Memory LLC v. NVIDIA Corp., 867 F.3d 1253, 1256 (Fed. Cir. 2017); Amdocs (Israel) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1303 (Fed. Cir. 2016).

[284] *Visual Memory*, 867 F.3d at 1255.

[285] *Id.* at 1257.

[286] *Id.* at 1261.

[287] *Amdocs*, 841 F.3d at 1303 (emphasis added).

[288] *Id.* at 1302.

[289] *Id.* at 1295.

[290] *Id.*

and argued that such an approach would conflate means-plus-function practice under § 112(f) with the § 101 analysis.[291] "That is not now the law, either in statute or in court decision," contended the majority in *Amdocs*.[292]

The same dispute about functional software claiming is also manifest at *Alice* step two, where the court must search for an inventive concept.[293] At step two, the issue is whether reciting a software function is sufficient to transform an otherwise abstract concept into a patentable application.[294] In *Berkheimer*, the claim recited "[a] method of archiving an item in a computer processing system" wherein object structures are stored "in the archive without substantial redundancy."[295] At the pleading stage, the Federal Circuit held there was a factual dispute as to whether "storing a reconciled object structure in the archive without substantial redundancy" represents an inventive concept.[296] In doing so, the court cited the specification's disclosure of a system that reduces redundancy by analyzing the variations between archived objects and linking common text and graphical elements.[297] Because the claims recite the function of storing objects "without substantial redundancy," the court found the claim language "capture[d] these improvements" described in the specification.[298]

This approach to *Alice* step two drew a sharp dissent from Judge Reyna in the denial of petition for *en banc* rehearing in *Berkheimer*.[299] In his dissent, Judge Reyna stressed that the search for an "inventive concept . . . is predominately a legal question focused on *the claims*."[300] That is, "the *claim* ha[s] to supply a 'new and useful' application of the idea in order to be patent eligible."[301]

[291] *Id.*

[292] *Id.*

[293] *See, e.g.*, Berkheimer v. HP Inc., 881 F.3d 1360, 1369–70 (Fed. Cir. 2018).

[294] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014).

[295] *Berkheimer*, 881 F.3d at 1366, 1370.

[296] *Id.* at 1370.

[297] *Id.* at 1369–70.

[298] *Id.* at 1370.

[299] Berkheimer v. HP Inc., 890 F.3d 1369, 1376 (Fed. Cir. 2018) (Reyna, J., dissenting).

[300] *Id.* at 1379 (internal quotation marks omitted).

[301] *Id.* (internal quotation marks omitted).

Although not stated explicitly, the dissent implies that a claim cannot "capture" a specific technological improvement just by stating its function.[302] Instead, the claim would likely need to recite the actual *means* for performing the claimed function.

Regardless of whether the issue is framed as an inquiry under step one or two, the basic disagreement centers on whether claims directed to software functions are inherently abstract, or if they can pass § 101 based on technological improvements described in the specification. On one hand, it makes sense that courts should look to the specification to determine what the invention covers. After all, terms like "programmable operational characteristic" are largely meaningless unless they are placed in context. Likewise, where the patent describes detailed algorithms for achieving an improvement in computer technology, it seems reasonable to rely on that description to determine whether the claim is abstract.

The problem, however, is that even if the specification provides detailed or groundbreaking algorithms to improve computer technology, the claims ultimately cover broad functions like "determin[ing] a type of data stored by said cache" and "storing a reconciled object structure in the archive without substantial redundancy."[303] These functions could be performed by any number of algorithms, and there is no guarantee the claim would be limited to what is disclosed in the specification.[304] To the contrary, claim terms with broad plain meanings are generally *not* limited to the disclosed embodiments.[305] As a result, the claims can still be asserted against totally different algorithms that perform the same functions.

Because of this dilemma, some practitioners have urged Congress to adopt a compromise solution based on § 112(f).[306] Under this proposal, courts would rely on the specification to determine whether a claimed software function is patent-eligible,

[302] *Id.*

[303] *Berkheimer*, 881 F.3d at 1370.

[304] Lemley, *supra* note 26, at 923 ("The function they perform may be simple or complex, broad or narrow, but in the modern world the patent claims listed above effectively cover any device that performs that function in any way.")

[305] Hill–Rom Servs., Inc. v. Stryker Corp., 755 F.3d 1367, 1372 (Fed. Cir. 2014).

[306] Stasa & Berry, *supra* note 46.

while also limiting the scope of functional software claims to the disclosed embodiments.[307] Although this idea makes sense in theory, there are practical hurdles to implementing this type of solution for software patents, particularly for Article III courts. The next section discusses this proposal as well as other legislative amendments aimed at providing more clarity to the § 101 analysis.

V. IMPLICATIONS FOR PROPOSED LEGISLATIVE REFORMS

With broad dissatisfaction at the current state of the law on patent eligibility, attention has once again turned to Congress for a legislative fix. Proponents of legislative reform argue that legislative action is required to overturn the Supreme Court's decisions in *Mayo* and *Alice*, which they believe to be recent distortions in patent law.[308] One patent bar association, for example, contend that legislative action is needed to "return the law to what the 1952 Patent Act meant to provide."[309]

In reality, redrafting the statutory definition of patent-eligible subject matter is an unprecedented step that would likely go beyond abrogating *Mayo* and *Alice*. The statutory language of § 101 has remained largely unchanged since the Patent Act of 1793, which established the categories of patentable subject matter.[310] Those categories were in turn derived from contemporaneous English standards for patent eligibility.[311] In short, the statutory definition of patentable subject matter has always been broadly drawn, and the standard has developed largely through common law.[312] If any of the leading proposals are enacted, it would be the first time that Congress intervened to dramatically broaden patentable subject matter.

[307] *Id.*

[308] AIPLA PROPOSAL, *supra* note 42, at 2, 10.

[309] *Id.* at 2.

[310] 1 Stat. 318, 319 § 1 (1793).

[311] In re Bilski, 545 F.3d 943, 968 (Fed. Cir. 2008) (Dyk, J., concurring).

[312] Federico, *supra* note 49, at 186. I use the term "common law" broadly to describe the development of legal standards through case-by-case adjudication. This includes judicial interpretations of ambiguous statutes as well as constitutional decisions. *See, e.g.*, Thomas W. Merrill, *The Common Law Powers of Federal Courts*, 52 U. CHI. L. REV. 1, 5 (1985).

That is not to say Congress should reject a legislative fix just because the statutory language has not been changed before. Proponents of reform correctly note that *Alice*'s two-step standard is nebulous and difficult to apply.[313] And it may also seem absurd that courts should rely on statutory language crafted in the early Industrial era to evaluate patents covering smart-phones and self-driving cars. Nevertheless, Congress must approach legislative reform with a clear understanding of why courts are invalidating so many software patents under the current statutory definition of patentable subject matter. As the previous section describes, a key question is whether software functions are inherently abstract.

This section will review the current leading proposed legislative amendments to § 101 and explain why they fail to address the underlying issue of functional software claims. It will also consider whether courts can narrowly construe functional software claims to cover only the specific algorithms or programming described in the specification, which is similar to the current approach under § 112(f). This section argues that although such a proposal might work in theory, it presents practical difficulties for courts because it requires guidelines on what types of algorithms and programs are definite enough to support the claimed functions.

A. *Proposed Legislative Amendments*

Proposals to amend § 101 have emerged from various bar and patent owner groups, including the American Bar Association (ABA) Section of IP law, the American Intellectual Property Law Association (AIPLA), and the Intellectual Property Owners Association (IPO).[314] Although the details might differ, the proposed

[313] AM. INTELL. PROP. L. ASS'N, *Joint AIPLA–IPO Proposal on Patent Eligibility*, at 2, https://www.aipla.org/docs/default-source/uploadedfiles/documents/resources2/reports/2017aipladirect/documents/aipla-report-on-101-reform-5-19-17-errata.pdf?sfvrsn=138c9ce7_1 (last accessed Aug. 3, 2018) [hereinafter JOINT AIPLA–IPO PROPOSAL] ("The Federal Circuit, the district courts, and the U.S. Patent and Trademark Office are all struggling to find a principled formula to guide their decision-making . . . .").

[314] *Id.*; Letter from Donna P. Suchy, Section Chair, American Bar Association Section of Intellectual Property Law, to the Hon. Michelle K. Lee, Director of the USPTO (Mar. 28, 2017), https://www.americanbar.org/content/dam/aba

amendments all aim to abrogate the two-step test established by *Mayo* and *Alice*. Likewise, most proposals attempt to carve out functional claiming considerations from the patent eligibility analysis.

In June 2018, Congress introduced the Restoring America's Leadership in Innovation Act of 2018, which incorporates the AIPLA-IPO joint proposal nearly verbatim.[315] AIPLA-IPO's joint proposal would add two sub-sections to § 101.[316] Section 101(b) specifies that an invention is ineligible "only if the claimed invention as a whole exists in nature independent of and prior to any human activity, or can be performed solely in the human mind."[317] Under the proposed amendment, any "result of human actions as applied to nature" would be patent-eligible.[318]

Proposed § 101(c) states that eligibility "shall be determined without regard to without regard to the requirements or conditions of sections 102, 103, and 112 of this title, the manner in which the claimed invention was made or discovered, or whether the claimed

/administrative/intellectual_property_law/advocacy/advocacy-20170328-comments.authcheckdam.pdf [hereinafter ABA Proposal].

[315] Restoring America's Leadership in Innovation Act of 2018, H.R. 6264, 115th Cong. (2d. Sess. 2018).

[316] Under the proposed IPO–AIPLA amendment, § 101 would read:

> (a) Whoever invents or discovers, and claims as an invention, any useful process, machine, manufacture, composition of matter, or any useful improvement thereof, shall be entitled to a patent therefor, subject only to the conditions and requirements set forth in this title.
>
> (b) A claimed invention is ineligible under subsection (a) if and only if the claimed invention as a whole (i) exists in nature independently of and prior to any human activity or (ii) is performed solely in the human mind.
>
> (c) The eligibility of a claimed invention under subsections (a) and (b) shall be determined without regard to: (i) the requirements or conditions of sections 102, 103, and 112 of this title; (ii) the manner in which the claimed invention was made or discovered; or (iii) whether the claimed invention includes an inventive concept.

JOINT AIPLA–IPO PROPOSAL, *supra* note 313, at 4.

[317] *Id.*

[318] *Id.*

invention includes an inventive concept."[319] Instead of relying on § 101 to invalidate patents, § 101(c) would "return the inquiry to well-developed legal principles" for patentability developed under other sections of the patent statute.[320] The goal is to "stop decision makers from confusing the patent eligibility inquiry with the enablement, written description, and definiteness inquiries under Section 112."[321] The AIPLA criticized the Supreme Court for "confus[ing] its own early cases, which repeatedly compare the scope of claims to the scope of patent disclosures to determine whether claims are in fact too broad, an inquiry required by the enablement and written description requirements."[322]

The AIPLA-IPO proposal represents a fairly radical change to patentable subject matter. Not only would it abrogate *Mayo* and *Alice*, but it also undercuts the reasoning in seminal nineteenth century cases like *Morse*. Because the amendment limits abstract ideas to activities "performed solely in the human mind," any computer implementation of an abstract idea, no matter how routine, would likely be patentable. After all, a hard-drive containing a program for hedging risk does not "exist in nature," nor can it be "performed solely in the human mind." Likewise, Samuel Morse's claim for using electric current to generate characters or signs at a remote location would also be patent-eligible, since this process cannot be performed solely in the human mind either.[323]

Although the AIPLA-IPO proposal is likely easier to apply than the current *Alice* two-step test, it does not resolve the underlying concern that software patents can be used to circumvent the judicial prohibition on patenting abstract ideas. In the modern world, few business processes or financial transactions are performed without computers. Accordingly, taking an abstract idea like intermediated settlement and reciting "apply it on the computer" does not impose any meaningful limitations to the abstract idea itself.[324] Simply stated, a patent that covers "intermediated settlement on a computer"

[319] *Id.*
[320] AIPLA PROPOSAL, *supra* note 42, at 2.
[321] *Id.* at 15.
[322] *Id.* at 16.
[323] O'Reilly v. Morse, 56 U.S. 62, 113 (1853).
[324] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2358 (2014).

has essentially the same scope as a patent on "intermediated settlement" generally. And contrary to the AIPLA-IPO's suggestion, it is far from clear that other patentability requirements could address this problem. After all, a claim on a revolutionary new type of intermediated settlement may be well-defined and readily implemented on a computer. Thus, the claim may very well satisfy the requirements of §§ 112, 102 and 103. Yet such a claim might still preempt a basic business practice.

Compared to the AIPLA-IPO joint proposal, the ABA's proposal to amend § 101 is more modest.[325] The ABA's proposal would add another subsection to § 101 under which a patent application may be "denied eligibility under this section 101 on the ground that the scope of the exclusive rights under such a claim would preempt the use by others of all practical applications of a law of nature, natural phenomenon, or abstract idea."[326] In a letter to the PTO Director, the ABA noted "the consistency with which the Court states that its fundamental concern is the potential 'preemption' of the use of building blocks like laws of nature, natural phenomenon and abstract ideas . . . . At its core, preemption is the

[325] ABA Proposal, *supra* note 314.

[326] *Id.* at 3. Under the ABA's proposed amendment, § 101 would read:

> a) Eligible Subject Matter – Whoever invents or discovers any useful process, machine, manufacture, or composition of matter, or any useful improvement thereof, shall be entitled to obtain a patent on such invention or discovery, absent a finding that one or more conditions or requirements under this title have not been met.
>
> b) Exception – A claim for a useful process, machine, manufacture, or composition of matter, or any useful improvement thereof, may be denied eligibility under this section 101 on the ground that the scope of the exclusive rights under such a claim would preempt the use by others of all practical applications of a law of nature, natural phenomenon, or abstract idea. Patent eligibility under this section shall not be negated when a practical application of a law of nature, natural phenomenon, or abstract idea is the subject matter of the claims upon consideration of those claims as a whole, whereby each and every limitation of the claims shall be fully considered and none ignored. Eligibility under this section 101 shall not be negated based on considerations of patentability as defined in Sections 102, 103 and 112, including whether the claims in whole or in part define an inventive concept.

Id. at 3–4.

driving force behind the Court's jurisprudence."[327] Unlike the AIPLA-IPO joint proposal, the ABA proposal would preserve the courts' ability to invalidate patents that are directed to a computer, but nevertheless preempt an abstract idea.

The ABA proposal, however, attempts to limit judicial authority over patentable subject matter in other respects. The proposed amendment requires courts and the PTO to "fully consider[] . . . each and every limitation of the claims."[328] And it prohibits courts from negating patent eligibility based on "consideration of those claims" as a whole.[329] This section seems to address the *Alice* two-step test, which requires the court to resolve what the claim is "directed to" as a whole.[330] Nevertheless, the amendment is unclear because it does not specify what patentable weight courts must assign to each limitation. Even under the current two-step standard, it would be unusual, and likely improper, for a court to ignore claim limitations entirely. For example, the patent in *Alice* implemented a method of intermediated settlement on a computer.[331] The Supreme Court held the claim was "directed to . . . a method of exchanging financial obligations . . . ."[332] In doing so, the Court still *considered* the computer implementation step, but found that it was not "enough for patent eligibility."[333] Accordingly, it is not immediately clear how the ABA proposal would actually alter the *Alice* step two inquiry.

Like the AIPLA-IPO proposal, the ABA's proposed amendment also tries to separate § 101 from other patentability requirements. The ABA's proposal states that "[e]ligibility under this section 101 shall not be negated based on considerations of patentability as defined in Sections 102, 103 and 112, including whether the claims in whole or in part define an inventive concept."[334] According to the ABA, "[t]he proposal at least substantially mitigates if not resolves

[327] *Id.* at 2.
[328] *Id.* at 3.
[329] *Id.*
[330] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014).
[331] *Id.* at 2358.
[332] *Id.* at 2356.
[333] *Id.* at 2350.
[334] ABA Proposal, *supra* note 314, at 3.

newly injected ambiguity and confusion into the eligibility determination caused by the use of factors that are relevant only to novelty, obviousness, and the requirements of particularity in claiming an invention."[335]

This represents a flawed approach to functional claims. Prohibiting courts from considering the requirements or conditions of § 112 presumes that courts are doing so in the first place. This presumption misunderstands why courts invalidate functional software claims. Courts do so because they consider software functions to be *inherently abstract*, not because the claims are unclear in scope or lack written description support. Under this view, a software function represents the idea of performing tasks on generic computer components without actually explaining how to do it. They are the digital equivalent of "a machine that applies Bernoulli's principle to fly." Although functional claims might also raise § 112 issues, that does not mean they are otherwise eligible under § 101. Therefore, even if Congress adopts the ABA's proposal, courts could continue to invalidate functional software claims under the standard set forth by *Bilski* and *Alice*.

B. *Extending the Legislative Compromise under Section 112 (f) to Patent Eligibility*

Ultimately, the AIPLA-IPO and ABA proposals do not resolve underlying disputes about functional software claims, nor do they address judicial concerns that such claims will hinder innovation. As an alternative, some have proposed tackling functional software claiming more directly by incorporating the requirements of § 112(f) into the patent eligibility determination.[336] Under this alternative approach, if a patent recites an ineligible software function, then courts will limit the claim to whatever software code or algorithms are disclosed in the embodiments.[337] As one commentator argues, "[p]reemption concerns [under Section 101] may be addressed

[335] *Id.*
[336] Stasa & Berry, *supra* note 46.
[337] *Id.*

adequately by disclosure-based limits on claim scope, rather than by precluding patentability *in toto*."[338]

One way to implement this proposal is through legislative amendment. For example, the existing language of § 112(f) could be appended to a new subsection under § 101. This subsection would state that "[a] claim directed to a judicially-recognized exception . . . shall be construed to cover the structures, materials, or acts described in the specification and equivalents thereof . . . ."[339] A legislative fix, however, may be unnecessary if courts take a more expansive approach of means-plus-function claims. Section 112(f) applies to claims that recite "a specified function without the recital of structure, material, or acts in support thereof."[340] This language could theoretically cover a wide swath of functional software claims that courts have found ineligible under § 101. Stated differently, if a claim would otherwise run afoul of § 101 because it is drafted in a purely functional manner, courts could find that such a claim invokes § 112(f) and narrowly construe the claim to the disclosed embodiments only.

The trend towards relaxing the standard for construing claims as means-plus-function terms already started with the Federal Circuit's decision in *Williamson*, which overturned the strong presumption that patentees only invoke § 112(f) if they use specific language like "means for."[341] Presently, however, the impact of *Williamson* is likely blunted by the courts' willingness to invalidate functional software claims under § 101 altogether without reaching the § 112(f) question. Courts could invoke § 112 to address judicial concerns about overbroad or vague software patents, instead of invalidating such claims under § 101.

There are several benefits to limiting functional software claims to the disclosed embodiments. First, narrowing functional software claim to a specific algorithm would make the claim less abstract. The Federal Circuit has reasoned that claims which are limited to a specific means for performing a claimed function are more likely to

[338] *Id.* (emphasis in original).
[339] *Id.*
[340] 35 U.S.C. § 112(f) (2011).
[341] Williamson v. Citrix Online, LLC, 792 F.3d 1339, 1359–62 (Fed. Cir. 2015).

be patent-eligible.[342] For example, in *Enfish, LLC v. Microsoft Corp.*, the Federal Circuit held that a claim directed to storing tabular data was patent-eligible in part because the claim expressly used means-plus-function language.[343] Thus, "the claims are not simply directed to *any* form of storing tabular data, but instead are specifically directed to a *self-referential* table for a computer database."[344] The patent did not involve just "general-purpose computer components [that] are added post-hoc to a fundamental economic practice or mathematical equation. Rather, the claims are directed to a specific implementation of a solution to a problem in the software arts. Accordingly, [the court found] the claims at issue are not directed to an abstract idea."[345]

Second, this proposed approach would address concerns about preemption that underpin judicial decisions on patent eligibility. In *Morse*, the Supreme Court invalidated Morse's patent for using electric current to generate characters or signs at a remote location because, for the purposes of infringement, "it matters not by what process or machinery the result is accomplished."[346] Likewise, the Court in *Alice* observed that allowing patentees to "claim any principle of the physical or social sciences by reciting a computer system configured to implement the relevant concept" would broadly preempt the future use of those concepts.[347] Limiting the claims to the algorithms and code described in the specification would obviate these concerns. In *Morse*, the patent would only cover the process and machines that Samuel Morse invented and disclosed.[348] If a subsequent inventor develops a better telegraph, they would not be liable for infringement. Likewise, the patent in *Alice* would only cover any specific algorithms for intermediated settlement described in the specification.

[342] Enfish, LLC v. Microsoft Corp., 822 F.3d 1327, 1335–36 (Fed. Cir. 2016); McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1314 (Fed. Cir. 2016).

[343] *Enfish*, 822 F.3d at 1335–36.

[344] *Id.* at 1337.

[345] *Id.* at 1339.

[346] O'Reilly v. Morse, 56 U.S. 62, 113 (1853).

[347] Alice Corp. v. CLS Bank Int'l, 134 S. Ct. 2347, 2359 (2014).

[348] *O'Reilly*, 56 U.S. at 124.

Third, limiting the scope of functional software claims would also resolve the competing judicial approaches to functional software claims. As described in Section IV.C, some cases have held software functions are inherently abstract, such that a patent reciting purely functional language will fail § 101 regardless of what the specification discloses.[349] By contrast, other cases look to the patent specification to determine if the claimed function is performed with patent-eligible technology.[350] The proposed approach represents a compromise between these two lines of cases. Courts and the PTO must rely on the specification to determine if the claim is eligible, and, therefore, will not invalidate a claim simply because it recites a function or end result. On the other hand, functional software claims will be restricted in scope and cannot cover algorithms for performing the claimed function that are not disclosed in the specification.

Despite these possible benefits, this approach may prove difficult for Article III courts to apply in practice. Section 112(f) is premised on the idea that courts can identify a corresponding structure or process in the specification to perform the claimed function. For software claims, the corresponding structure or process is usually an algorithm or software object.[351] The problem, however, is that algorithms and software objects are essentially logical procedures for performing tasks.[352] At bottom, they are still a series of functions. As Professor Kevin Collins observes, software "[a]lgorithms are recursive entities: algorithms have sub-algorithms, which have sub-sub-algorithms, etc."[353] In short, "*software is functional all the way down*."[354]

Given the recursive nature of software, courts must find a level of abstraction that satisfies § 101 and § 112(f). In other words, courts must "identify a bottom as a matter of policy—a level of generality below which a functional property of a software program

[349] *See supra* Section IV.C.
[350] *Id.*
[351] Collins, *supra* note 24, at 1455–56.
[352] *Id.*
[353] *Id.* at 1463.
[354] *Id.* at 1464 (emphasis added).

counts as metaphorical structure . . . ."[355] This should be possible in theory. According to Professor Mark Lemley, there are "well-understood class[es] of software objects" that could serve as the metaphorical "structure."[356] He further notes that the same theoretical problem exists in more conventional claims. That is, "[a] jackhammer functions too, but we have no trouble distinguishing the function it performs from the way in which it performs that function. The same can be said of software."[357]

Although it is likely possible to identify an "acceptable" level of abstraction for software functions, it remains an open question as to whether courts are institutionally equipped to do so. Professor Collins expressed doubt that an Article III court is capable of resolving this question with any consistency.[358] He notes this inquiry would "require consultation with computer scientists to create a taxonomy of a variety of levels of abstraction at which the functional properties of a software program can be formulated."[359] And it would also require courts "to identify the level of abstraction at which algorithmic descriptions of software become sufficiently specific to count as the descriptions of the metaphorical structure of software inventions."[360] From an institutional standpoint, courts have questioned their own ability to resolve this type of policy question. In *Benson*, the Supreme Court stated that "[t]he technological problems tendered in the many briefs before us indicate to us that considered action by the Congress is needed."[361] The Court in *Flook* noted how Congress should rely on empirical data to resolve "[d]ifficult questions of policy concerning the kinds of programs that may be appropriate for patent protection and the form and duration."[362]

From an institutional standpoint, the PTO is likely better suited to establish guidelines on what types of software algorithms connote

[355] *Id.* at 1466.
[356] Lemley, *supra* note 26, at 960.
[357] *Id.*
[358] Collins, *supra* note 24, at 1466.
[359] *Id.*
[360] *Id.*
[361] Gottschalk v. Benson, 409 U.S. 63, 73 (1972) (footnote omitted).
[362] Parker v. Flook, 437 U.S. 584, 595 (1978).

structure and which are merely functions.[363] The PTO already employs thousands of employees, many of whom possess relevant scientific and technical training.[364] Likewise, the agency has divisions dedicated to economic research and analysis that could evaluate the impact of whatever guidelines it develops regarding software patents.[365] While giving this problem to the PTO seems sensible, the agency currently lacks legal authority to interpret § 101.[366] As a result, the PTO cannot promulgate legally binding rules on what types of software algorithms or functions are sufficiently concrete.[367] Whether the PTO should be given substantive rulemaking authority over § 101 remains a contentious topic and is beyond the scope of this article.[368] Opponents of giving substantive rulemaking authority to the PTO express concerns about institutional competence and agency capture.[369] But at least for finding a workable approach to functional software claims, the PTO is likely best suited to resolve the proper level of abstraction for software.

[363] Golden, *supra* note 262, at 1096–97. As Professor John Golden argues:

> [T]he malleability of technology and of approaches to describing it will often reveal statutory ambiguity with respect to subject–matter eligibility that traditional techniques of statutory interpretation cannot satisfactorily resolve. The categorical, policy–laden nature of the resulting interpretive questions suggests that these questions are better left to primary resolution by a policy organ specially concerned with such questions–namely, an administrative agency.

Id. at 1096.

[364] *Id.* at 1098 ("[T]wo of the central challenges for subject–matter eligibility involve formulating doctrine (1) that the USPTO's thousands of examiners can effectively administer and (2) that can provide effective guidance to patent attorneys, patent agents, and their clients . . . .").

[365] *Office of the Chief Economist*, USPTO, https://www.uspto.gov/about-us/organizational-offices/office-policy-and-international-affairs/office-chief-economist (last visited Aug. 10, 2018).

[366] Tafas v. Doll, 559 F.3d 1345, 1352 (Fed. Cir. 2009), *vacated en banc on other grounds*, 328 F. App'x 658 (Fed. Cir. 2009), *dismissed as moot*, 586 F.3d 1369 (2009).

[367] *Id.*

[368] Golden, *supra* note 262, at 1095–1111.

[369] *Id.*

VI. CONCLUSION

Nearly two centuries of judicial decisions have expressed policy concerns about functional claims. And while Congress tried to resolve this issue in the 1952 Patent Act, the advent of general-purpose computers raised new and unforeseen questions about functional claiming that courts are still grappling with. After *Alice*, the preferred solution among many judges is to invalidate functional software claims altogether. Reasonable minds might differ on whether this approach is a faithful application of § 101 and § 112. Nevertheless, these decisions reflect legitimate policy concerns about whether patentees can use functional software claiming to improperly tie up abstract ideas. Legislative action to address patentable subject matter should not ignore these concerns. And yet, the current set of proposed amendments to § 101 would dramatically limit judicial discretion without resolving questions about the preemptive effects of software patents. Moreover, by rigidifying the patent eligibility standard, the proposed legislative reforms could leave courts and the PTO less capable of contending with issues raised by new and unforeseen technological developments.