ALAPPAT REDUX: SUPPORT FOR FUNCTIONAL LANGUAGE IN SOFTWARE PATENT CLAIMS

*Andrew Chin**

ABSTRACT

The Federal Circuit has suggested in some recent cases that any algorithm can serve as adequate structural support for a means-plus-function element in a software patent claim under § 112(f). A recent proposal by Mark Lemley fully endorses this proposition and seeks its broader application. The concept of an algorithm, however, is too slippery to serve as the basis for such a rule. In this Article, I argue that this overreliance on the algorithm concept originated in a revisionist gloss on the Federal Circuit's 1994 Alappat *decision. Informed by a closer reading of what* Alappat *actually has to say about claim construction under § 112(f), I propose a more stable "concrete causation" standard that is not only applicable to all technologies, but also well-aligned with the reforms in the software field intended by Lemley's proposal.*

I. RECENT SCRUTINY OF FUNCTIONAL CLAIMING

THE alleged systemic failure of software patent claims to provide notice of their scope has received considerable attention in recent years. According to the authors of *Patent Failure: How Judges, Bureaucrats, and Lawyers Put Innovators at Risk*, "the abstractness of software technology inherently makes it more difficult to place limits on abstract claims in software patents."[1] To characterize all software technology as problematically abstract, however, is to paint it with too broad a brush.

As I argued in a 2009 symposium volume on *Patent Failure*, even the book's examples of allegedly abstract patent claims are amenable to reasonably straightforward determinations of claim scope.[2] To be more specific, abstractions in most of these example–claims no longer appear to be

* Associate Professor, University of North Carolina School of Law. The author wishes to thank Keith Robinson, Simone Rose, and Hilda Galvan for helpful discussions; and the organizers and editors for their exemplary efforts in producing an outstanding symposium program.

1. JAMES BESSEN & MICHAEL J. MEURER, PATENT FAILURE: HOW JUDGES, BUREAUCRATS, AND LAWYERS PUT INNOVATORS AT RISK 201 (2008).

2. *See* Andrew Chin, *On Abstraction and Equivalence in Software Patent Doctrine: A Response to Bessen, Meurer and Klemens*, 16 J. INTELL. PROP. L. 197, 200 (2009).

problematic once they are properly interpreted as "means-plus-function" or "step-plus-function" elements[3] in accordance with § 112(f) of the Patent Act.[4] Under this statutory provision, patent claims that use functional language to describe kinds of structures, materials, or acts are not to be construed as encompassing all such means or steps capable of performing the described function; instead, they should be interpreted narrowly to cover only the particular means or steps that are described in the patent specification and equivalents thereof.[5]

This rule of narrow construction, in which abstract functional claim language is replaced with concrete particular structures, materials, or acts, is a potentially powerful safeguard against overbreadth and uncertainty in the scope of software patent claims. In a recent article, *Software Patents and the Return of Functional Claiming*, Mark Lemley has gone so far as to argue that a more assiduous application of § 112(f) to functional language in software patent claims might be enough to address most contemporary objections to software patents "with one fell swoop."[6]

I applaud Lemley's call for means-plus-function and step-plus-function claim elements to be faithfully identified and narrowly construed in accordance with statute. In this Article, however, I will argue that the application of § 112(f) brings into play potential ambiguities in the specification of supporting structures, materials, and acts. The pressing need to resolve these ambiguities in the wake of *Bilski v. Kappos*[7] calls for a careful review of the Federal Circuit's case law on the construction of software means-plus-function and step-plus-function claims. In particular, a 1994 Federal Circuit en banc decision, *In re Alappat*,[8] warrants prominent consideration as a § 112(f) precedent. While *Alappat* is better known in the annals of software patent law as a once-important precedent on the patent eligibility of programmed computers under § 101,[9] a close reading of the decision shows that its force in the § 112(f) context

---

3. *See id.* at 224.

4. Section 112(f) provides:

> 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.

35 U.S.C. § 112(f) (2006 & Supp. V 2011).
Section 112(f) is the 2011 recodification of § 112, ¶ 6; the newer designation will be used to refer to the provision throughout this Article.

5. *See id.*

6. Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 Wisc. L. Rev. 905, 909.

7. 130 S. Ct. 3218 (2010).

8. 33 F.3d 1526 (Fed. Cir. 1994) (en banc), *abrogated by In re* Bilski, 545 F.3d 93 (Fed. Cir. 2008).

9. A plurality of the Federal Circuit has recently questioned *Alappat*'s continuing relevance in the patent-eligibility context. *See* CLS Bank Int'l v. Alice Corp., 717 F.3d 1269, 1292 (Fed. Cir. 2013) (en banc) (Lourie, J. concurring) (per curiam) ("Not only has the world of technology changed, but the legal world has changed. The Supreme Court has spoken since *Alappat* on the question of patent eligibility, and we must take note of that change.").

remains undiminished, warranting an even narrower construction of many software claims than that urged by Lemley.

This Article's conclusions are immediately applicable to questions recently posed by the U.S. Patent and Trademark Office's recently created Partnership for Enhancement of Quality of Software Related Patents ("Software Partnership").[10] In February 2013, Patent Office officials convened public discussions in Silicon Valley and New York City focused primarily on the topic of establishing clear boundaries for software related patent claims that use functional language.[11] At the heart of the agency's inquiries was the sufficiency of structural disclosure to support and define the scope of claims that recite functional language.[12] Several commenters at the discussions noted parallels between the Patent Office's software initiative and the public comment process that accompanied the agency's development of the 2001 Utility Examination Guidelines,[13] specifically addressing the patentability of DNA molecules.[14] It therefore appears to many observers that the Patent Office is preparing to take up Lemley's proposal even before the article's journal

---

10. Request for Comments and Notice of Roundtable Events for Partnership for Enhancement of Quality of Software-Related Patents, 78 Fed. Reg. 292–93 (Jan. 3, 2013).

11. *See id.* at 294. Specifically, most presenters (including the author) addressed the following questions:

> 1. When means-plus-function style claiming under 35 U.S.C. 112(f) is used in software-related claims, indefinite claims can be divided into two distinct groups: claims where the specification discloses no corresponding structure; and claims where the specification discloses structure but that structure is inadequate. In order to specify adequate structure and comply with 35 U.S.C. 112(b), an algorithm must be expressed in sufficient detail to provide means to accomplish the claimed function. In general, are the requirements of 35 U.S.C. 112(b) for providing corresponding structure to perform the claimed function typically being complied with by applicants and are such requirements being applied properly during examination? In particular:
>
> (a) Do supporting disclosures adequately define any structure corresponding to the claimed function?
>
> (b) If some structure is provided, what should constitute sufficient 'structural' support?
>
> (c) What level of detail of algorithm should be required to meet the sufficient structure requirement?
>
> 2. In software-related claims that do not invoke 35 U.S.C. 112(f) but do recite functional language, what would constitute sufficient definiteness under 35 U.S.C. 112(b) in order for the claim boundaries to be clear? In particular:
>
> (a) Is it necessary for the claim element to also recite structure sufficiently specific for performing the function?
>
> (b) If not, what structural disclosure is necessary in the specification to clearly link that structure to the recited function and to ensure that the bounds of the invention are sufficiently demarcated?
>
> 3. Should claims that recite a computer for performing certain functions or configured to perform certain functions be treated as invoking 35 U.S.C. 112(f) although the elements are not set forth in conventional means-plus-function format?

See id.

12. *See id.*

13. *See* Utility Examination Guidelines, 66 Fed. Reg. 1092 (Jan. 5, 2001).

14. *See* J. Timothy Meigs, *Biotechnology Patent Prosecution in View of PTO's Utility Examination Guidelines*, 83 J. PAT. & TRADEMARK OFF. SOC'Y 451, 458–68 (2001) (describing development of guidelines by the Patent Office).

publication[15] and looking to § 112(f) as the principal policy lever to curb perceived overbreadth and indefiniteness due to abstraction in software patent claims.

If and when the Patent Office develops guidelines with the aim of promulgating a robust legal standard for supporting "structure" under § 112(f), the devil will be in the details. As this Article will explain, a careful reading of the Federal Circuit's *Alappat* decision can provide guidance to the agency on this increasingly pivotal issue in the examination of software patent claims.

The remainder of this Article is organized as follows. Section II describes Lemley's proposal in the context of previous commentary and relevant caselaw. Section III explains the *Alappat* majority's approach to claim construction under § 112(f) and points out the court's divergence from both Lemley's interpretation and a gloss offered by the Federal Circuit in its subsequent *WMS Gaming* decision.[16] Drawing from *Alappat*'s insights, Section IV proposes, in the spirit of a friendly amendment to Lemley's proposal, a "concrete causation" standard for supporting "structure" under § 112(f). Section V concludes with additional arguments in support of my proposal.

II. LEMLEY'S PROPOSAL

One of Bessen and Meurer's examples of a problematically abstract software patent was a 1988 patent granted to AT&T Bell Labs's Narendra Karmarkar for improved "[m]ethods and apparatus for efficient resource allocation."[17] Karmarkar's patent specification discloses, *inter alia*, a new algorithm for solving linear programming problems (a fundamental computational problem in operations research) in considerably less time than previous methods.[18] As I described in my 2009 response to Bessen and Meurer, Karmarkar's breakthrough involved the use of a novel "interior-point" projective transformation to search iteratively for solutions to a given problem, in contrast with prior art "exterior-point" methods.[19] I noted that nineteen of Karmarkar's twenty-two independent claims contained functional language, such as "normalizing," "selecting," and "transforming," without reciting supporting structure, material, or acts.[20]

15. *See, e.g.*, John R. Harris, *Patent Office to Explore Ways to Improve Quality of Software Patents*, MMM TECH L. & BUS. REP., http://www.mmmtechlaw.com/?p=6825 (last visited Sept. 1, 2013) (describing Topic 1 as "stem[ming] from" Lemley, *supra* note 6); Pamela Jones, *Groklaw's Response to the USPTO on the First Topic* 2 & n.1 (Feb. 10, 2013), http://www.uspto.gov/patents/law/comments/sw-e_groklaw_topic1_20130210.pdf (citing Lemley, *supra* note 6); Lisa Larrimore Ouellette, *Patent Troll Panel at Yale Law School*, PATENTLY-O (May 14, 2013), http://www.patentlyo.com/patent/2013/05/patent-troll-panel-at-yale-law-school.html (noting citation of Lemley, *supra* note 6 in presentation by USPTO Deputy Solicitor Nathan Kelley).

16. WMS Gaming, Inc. v. Int'l Game Tech., 184 F.3d 1339 (Fed. Cir. 1999).

17. U.S. Patent No. 4,744,028 (filed Apr. 19, 1985) (issued May 10, 1988).

18. *See id.*

19. *See* Chin, *supra* note 2, at 214–23 (describing Karmarkar's contributions).

20. *Id.*

Therefore, it should be construed narrowly as means-plus-function claims or step-plus-function claims in accordance with § 112(f).[21] Accordingly, I concluded that these claims should "be limited in scope to algorithms that implement a projective transformation as described in the specification."[22]

It is possible to read Karmarkar's claims more broadly. While § 112(f) of the Patent Act by its terms is applicable whenever an element of a claim is described in functional terms "without the recital of structure, material, or acts in support thereof,"[23] the Federal Circuit has tended to limit application of the provision to cases where "in the selection of claim language, the patentee must be taken to have exercised that option."[24] Thus, for Chief Judge Rader at least, the applicability of § 112(f) does not turn on the sufficiency or insufficiency of supporting structure for a functionally described element; the correct inquiry is whether the claim was drafted "advisedly to invoke the statutory mandates for means-plus-function clauses."[25]

Since practitioners generally prefer to use the term "means" to signify an intention to draft a claim that falls under § 112(f),[26] a court may be reluctant to construe a claim element as a means-plus-function clause if it does not expressly recite the term "means."[27] Similarly, the absence of the phrase "step for" in a process claim may raise a judicial presumption against a step-plus-function construction.[28] In contrast, in my 2009 article, I called for the application of § 112(f) to several of Karmarkar's independent process claims that—in my view—contain functional language without reciting supporting structure, material, or acts, even though there is no "step for" phrase to signify intent to draft in step-plus-function format.[29]

Along similar lines, Lemley describes his proposed approach to the application of § 112(f) in terms of textual fidelity to the statute:

> We don't need to change the statute to achieve this result. We don't even need to overrule existing cases. We just need to take seriously law that is on the books but doesn't seem to get applied in

21. *See id.* at 224. § 112, ¶ 6 was recodified as § 112(f) in 2011.
22. *See id.*
23. 35 U.S.C. § 112(f).
24. *See* Greenberg v. Ethicon Endo-Surgery, Inc., 91 F.3d 1580, 1584 (Fed. Cir. 1996).
25. *See* Cole v. Kimberly-Clark Corp., 102 F.3d 524, 533 (Fed. Cir. 1996) (Rader, J., dissenting) (citing York Prods. Inc. v. Cent. Tractor Farm & Family Ctr., 99 F.3d 1568, 1574 (Fed. Cir. 1996)).
26. *See* Robert C. Faber, Faber on Mechanics of Patent Claim Drafting § 3:29.8, at 3-113 (2009) ("To be sure you are under section 112, paragraph 6, use the pure 'means for . . . .' Other words lead to ambiguity and the need for a court to decide.").
27. *See, e.g.*, *Greenberg*, 91 F.3d at 1584 ("Nonetheless, the use of the term 'means' has come to be so closely associated with 'means-plus-function' claiming that it is fair to say that the use of the term 'means' . . . generally invokes section 112(6) and that the use of a different formulation generally does not.").
28. *See* Generation II Orthotics, Inc. v. Med. Tech., Inc., 263 F.3d 1356, 1368 (Fed. Cir. 2001).
29. *Compare* Chin, *supra* note 2, at 225–26, *with* U.S. Patent No. 4,744,028, cls. 1, 3, 15, 16, 18, 20, 21, 23, 26, 27, 33, 35, 36 (filed Apr. 19, 1985).

> practice. The Federal Circuit or the Supreme Court could, with one fell swoop, do away with most of the problem of overclaiming in software patents—and with it, most of the problems with software patents. All it needs to do is to take the statute at face value and limit functional claims to the particular way the patentee implemented that function.[30]

Though Lemley does not so state, his proposal would dispense with the issue of whether the patentee intended to draft a claim subject to the provisions of § 112(f). For Lemley, every patent claim that uses functional language must be limited in scope "to the actual structure, material, or acts the patentee built or described,"[31] whether or not it includes the "means for" or "step for" phrase as a signifier of the claim drafter's intent.[32]

Lemley also explains what his proposal would entail in the case of a patent claim directed to a general-purpose computer programmed to perform a sequence of functional steps. According to Lemley, in such a claim, "the 'structure' or 'acts' that perform the [recited] function are not simply 'a computer' but 'a computer programmed in a particular way.'"[33] Thus, the scope of such a claim should be "limited to that particular computer program and ones that work in the same way to achieve the same result."[34] In other words, "patentees will have to disclose the algorithms they use to achieve particular ends, and the patent will be limited to those algorithms and equivalents thereof."[35]

While Lemley argues persuasively that the functional elements of software patent claims should be construed narrowly in accordance with disclosures in the accompanying patent specification, his proposal to focus on disclosed algorithms does not fully address the Patent Office's call for standards regarding the sufficiency of structural disclosure. This is because the question of which "algorithms [are] use[d] to achieve particular ends" is so often an indeterminate inquiry.[36]

The term "algorithm" can refer to any "finite sequence of steps" that accomplishes a given task.[37] As an algorithm is usually described in computer science literature, it is common for some or all of the "steps" themselves to be tasks that can be decomposed further into sequences of more basic steps. A computer system thereby typically involves numerous "abstraction layers," with each successive, more abstract layer implementing its own set of functions through various algorithms comprising sequences

30. Lemley, *supra* note 6, at 947–48.
31. *Id.* at 947.
32. *See id.* at 944–46.
33. *See id.* at 947.
34. *See id.* at 948.
35. *Id.* at 947.
36. *Id.*
37. *See* MICROSOFT COMPUTER DICTIONARY 19 (1999) (defining "algorithm" as a "finite sequence of steps for solving a logical or mathematical problem or performing a task").

of functions previously implemented by the more concrete layers below.[38] To make matters even more complicated, abstraction layers often provide multiple distinct implementations and interpretations of a single function, using a versatile programming technique known as "indirection."[39] For example, the FreeBSD operating system uses indirection to implement a single "read system call" operation on disparate file-system organizations such as those in PC hard drives, CD-ROMs, and USB sticks.[40]

In sum, because of this multilayered, multiply interpreted approach to implementation, there is usually no single algorithm that can be said to constitute the "structure, material, or acts"[41] that support the functionality of a software-related invention. Lemley's proposal that the scope of functional elements in software patent claims "be limited to [disclosed] algorithms and equivalents thereof" therefore does not provide adequate guidance to the Patent Office or the courts for construing means-plus-function claims under § 112(f).[42]

The Federal Circuit actually stated a rule along the lines of Lemley's proposal in its 1999 *WMS Gaming* decision.[43] In that case, the district court construed a "means for assigning" limitation to cover "any table, formula or algorithm" for performing the functional assignment in question.[44] The Federal Circuit reversed, concluding that the district court "erred by failing to limit the claim to the algorithm disclosed in the specification."[45] The appeals court held that "[i]n a means-plus-function claim in which the disclosed structure is a computer, or microprocessor, programmed to carry out an algorithm, the disclosed structure is not the general purpose computer, but rather the special purpose computer programmed to perform the disclosed algorithm."[46] Accordingly, the "means for assigning" limitation had to be construed narrowly according to the algorithm illustrated in the specification.[47]

Lemley's article praises the post-*Bilski* progeny of *WMS Gaming*,[48]

---

38. *See* ANDREW S. TANENBAUM, STRUCTURED COMPUTER ORGANIZATION 2–4 (1976).

39. *See* Diomidis Spinellis, *Another Level of Indirection*, *in* BEAUTIFUL CODE: LEADING PROGRAMMERS EXPLAIN HOW THEY THINK 279–91 (Andy Oram & Greg Wilson, eds. 2007). Indirection is such a versatile approach to abstracting away implementation details that the claim that "[a]ll problems in computer science can be solved with another layer of indirection" has become a well-known aphorism among programmers. *See id.* at 279.

40. *See id.* at 279–82.

41. 35 U.S.C. § 112(f) (2006 & Supp. V 2011).

42. *See* Lemley, *supra* note 6.

43. WMS Gaming, Inc. v. Int'l Game Tech., 184 F.3d 1339 (Fed. Cir. 1999).

44. *See id.* at 1348.

45. *Id.*

46. *Id.* at 1349 (citing *In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994)).

47. *See id.*

48. *See* Lemley, *supra* note 6, at 954 (concluding that functional software claims that fail to disclose a supporting algorithm "are (and should be) invalid under *Aristocrat*"); *id.* at 926 & n.88 (noting with approval that "the Federal Circuit has of late been quite vigilant in limiting software patentees who write claims in means-plus-function format to the particular algorithms that implement those claims," and citing *Aristocrat* and other cases); id. at 949 & n.183 (same).

particularly the Federal Circuit's 2008 *Aristocrat* decision.[49] In *Aristocrat*, the patentee conceded that the only disclosed structural support for a recited means-plus-function limitation was "any standard microprocessor base [sic] gaming machine [with] appropriate programming."[50] The Federal Circuit cited a restatement of the *WMS Gaming* holding that "the corresponding structure for a § 112[(f)] claim for a computer-implemented function is the algorithm disclosed in the specification."[51] Finding "no algorithm at all disclosed in the specification" of the *Aristocrat* patent, the court concluded that the means-plus-function limitation lacked sufficient structural support and was therefore invalid for indefiniteness.[52]

Lemley's proposal is grounded in his concern that the Federal Circuit has not strictly followed *Aristocrat* in requiring the disclosure of an algorithm as structural support for a recited function. He notes that the Federal Circuit has sometimes been satisfied with finding structural support in "fairly general language rather than a specific implementation."[53] Moreover, when claim drafters have avoided explicitly using the term "means" by claiming structures capable of performing a recited function such as "a computer," "a processor," or "the Internet," the Federal Circuit has tended not to apply § 112(f) at all, effectively conferring "control over the claimed function" regardless of the disclosure or nondisclosure of structural support.[54]

In the closing paragraphs of its decision, the *Alappat* majority famously reasoned that "a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software."[55] Interestingly, this logic underpins both the holding of *WMS Gaming* and (according to Lemley) the Federal Circuit's failure to apply § 112(f) to functional elements where computing structures are recited. The *WMS Gaming* court inferred from *Alappat* that "[t]he structure of a microprocessor programmed to carry out an algorithm is limited by the disclosed algorithm," insofar as "[t]he instructions of the software program that carry out the algorithm electrically change the general purpose computer by creating electrical paths within the device."[56] Lemley, however, argues that "[b]y concluding that a general-purpose computer was a new machine whenever it was programmed with new instructions, the Federal Circuit

---

49. Aristocrat Techs. Austl. Pty. Ltd. V. Int'l Game Tech., 521 F.3d 1328 (Fed. Cir. 2008).
50. *See id.* at 1333 (alterations in original).
51. *See id.* (citing Harris Corp. v. Ericsson Inc., 417 F.3d 1241, 1249 (Fed. Cir. 2005)).
52. *See id.* at 1337–38.
53. Lemley, *supra* note 6, at 949 & n.184 (citing Typhoon Techs. v. Dell, Inc., 659 F.3d 1376 (Fed. Cir. 2011); *In re* Katz Interactive Call Processing Litig., 639 F.3d 1303 (Fed. Cir. 2011)).
54. *See* Lemley, *supra* note 6, at 927 & n.89 (citing Inventio AG v. ThyssenKrupp Elevator Ams. Corp., 649 F.3d 1350, 1359–60 (Fed. Cir. 2011)).
55. *In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994).
56. *See* WMS Gaming Ins. v. Int'l Game Tech., 184 F.3d at 1339, 1348 (citing *Alappat*, 33 F.3d at 1545).

opened the door to treating a programmed computer as physical structure rather than as a functional claim that had to be interpreted under section 112(f)."[57]

Overlooked in this too-pat reading of *Alappat* is the fact that the court's construction of the claims in question under § 112(f) did not involve the essentially metaphysical proposition that programming creates a new machine—i.e., a proposition concerning the § 101 subject-matter eligibility of a programmed computer that *CLS Bank* has recently called into doubt.[58] I will now discuss what *Alappat* actually had to say on the subject of claim construction under § 112(f).

III. READING *ALAPPAT* AS A § 112 PRECEDENT

In *Alappat*, the Federal Circuit reviewed the Patent Office's rejection of five claims, four of which were dependent from the first.[59] The representative claim read:

> 15. A rasterizer for converting vectors in a data list representing sample magnitudes of an input waveform into anti-aliased pixel illumination intensity data to be displayed on a display means comprising:
>
> (a) means for determining a vertical distance between the endpoints of each of the vectors in the data list;
>
> (b) means for determining an elevation of a row of pixels that is spanned by the vector;
>
> (c) means for normalizing the vertical distance and elevation; and
>
> (d) means for outputting illumination intensity data as a predetermined function of the normalized vertical distance and elevation.[60]

Construing this claim in accordance with § 112(f), the court replaced each of the four "means" terms in clauses (a)–(d) with what it determined to be the corresponding structure disclosed in the specification:

> 15. A rasterizer [a "machine"] for converting vectors in a data list representing sample magnitudes of an input waveform into anti-aliased pixel illumination intensity data to be displayed on a display means comprising:
>
> (a) [an arithmetic logic *circuit* configured to perform an absolute value function, or an equivalent thereof] for determining a vertical distance between the endpoints of each of the vectors in the data list;
>
> (b) [an arithmetic logic *circuit* configured to perform an absolute value function, or an equivalent thereof] for determin-

57. *See* Lemley, *supra* note 6, at 927–28 n.89 (citing *Alappat*, 33 F.3d at 1545).

58. *See* CLS Bank Int'l v. Alice Corp., 717 F.3d 1269, 1273–74 (Fed. Cir. 2013) (en banc) (Lourie, J. concurring) (per curiam).

59. *Alappat*, 33 F.3d at 1538–39.

60. *Id.*

> ing an elevation of a row of pixels that is spanned by the vector;
>
> (c) [a pair of *barrel shifters*, or equivalents thereof] for normalizing the vertical distance and elevation; and
>
> (d) [a *read only memory (ROM)* containing illumination intensity data, or an equivalent thereof] for outputting illumination intensity data as a predetermined function of the normalized vertical distance and elevation.[61]

Having construed Claim 15 narrowly in accordance with these structural limitations, the court reasoned that the claim "unquestionably recites a machine, or apparatus, made up of a combination of known electronic circuitry elements."[62] Observing that a "machine" is explicitly recognized as patent-eligible subject matter under § 101, the court proceeded to use the conclusion from its § 112(f) analysis—that Claim 15 recites a machine—as the starting point for its § 101 analysis.[63]

On the other hand, the *Alappat* court's § 112(f) claim-construction analysis is both rhetorically and logically unconnected to the court's later observation that a general-purpose computer, once programmed, becomes a new machine. The latter observation was a response to the patentee's admission that Claim 15 would read on a programmed general purpose computer, not a finding by the court concerning the scope of Claim 15.[64] To summarize, the *Alappat* majority's § 112(f) analysis informs its § 101 analysis, not—as *WMS Gaming*'s and Lemley's glosses[65] suggest—the other way around.

Thus, the *WMS Gaming* court's description of *Alappat*'s § 112(f) analysis as based on a finding that "[t]he instructions of the software program that carry out the algorithm electrically change the general purpose computer by creating electrical paths within the device,"[66] is simply revisionism. The *Alappat* majority made no mention of "electrical paths" being created through programming. Its § 112(f) analysis was instead appropriately grounded in the structural nature of the disclosed elements that it determined to correspond to each of the claimed "means" terms: arithmetic logic circuits, a barrel shifter, and a read-only memory.[67]

A close examination of *Alappat*'s patent specification also illuminates what (I have suggested elsewhere[68]) is the sine qua non of a structural element: its involvement in a causal process. As Table 1 illustrates, *Alappat* discloses several explicitly causal processes that together produce the

61. *Id.* at 1541.
62. *Id.*
63. *Id.* at 1541–42.
64. *See id.* at 1544–45.
65. *See supra* text accompanying notes 55–57.
66. *See* WMS Gaming Inc. v. Int'l Game Tech., 184 F.3d 1339, 1348 (Fed. Cir. 1999) (citing *Alappat*, 33 F.3d at 1545).
67. *See Alappat*, 33 F.3d at 1541.
68. *See* Andrew Chin, *The Ontological Function of the Patent Document*, 74 U. PITT. L. REV. 57–59 (2013) (discussing Centricut, LLC v. Esab Grp., 390 F.3d 1361 (Fed. Cir. 2004)).

functions of the claimed machine, including processes respectively involving the disclosed arithmetic logic circuit (the "ALU"), barrel shifters, and the read-only memory.

Disclosed Element	**Disclosed Causal Process**
arithmetic logic circuit	"[V]arious operations of rasterizer 40 . . . are timed by clock signals produced by a state machine in accordance with control data . . . One signal is a 'pixel clock' signal that is asserted to *cause* the rasterizer to receive each new vector list data element. . . . This [ALU] value is stored in a register 76 on the next pixel clock cycle."[69]
barrel shifter	"[P]riority encoder 86 *causes* barrel shifter 84 to shift its input to the left by the number of bits required . . . ."[70]
read-only memory	"The 8-bit intensity data stored in register 90 addresses a read only memory (ROM) 92 and *causes* ROM 92 to read out a 4-bit intensity data value which is stored in a register 94 on the next pixel clock cycle."[71]

Table 1. Causal processes involving each of the disclosed structural elements supporting Claim 15.

IV. A FRIENDLY AMENDMENT

As I have noted, *WMS Gaming*'s gloss on *Alappat* implicitly appeals to the view that an alteration in "electrical paths" amounts to a structural change;[72] however, such a view is not necessarily intuitive.[73] In my view, the reason such changes in the flow of electrons are cognizable as structural support for a claimed "means" is that they are the effects of causal processes.

Of course, the electrons that animate Alappat's rasterizer and today's computers are creatures of theoretical physics; they have never been directly observed.[74] Patentees cannot exhibit infringing electrons at trial. Yet it is routine, and not at all problematic, for patent claims to refer to electrons. This is because the theory of the electron is so well-developed and widely-accepted that scientists can make and use electricity and other observable effects of electrons without worrying about whether electrons

69. U.S. Patent No. 5,440,676 cols.3–4 (filed Jan. 29, 1988) (emphasis added).

70. *Id.* at col.6 (emphasis added).

71. *Id.* at cols.6–7 (emphasis added).

72. *See supra* text accompanying note 56.

73. *Cf. In re* Nuijten, 500 F.3d 1346, 1357 (Fed. Cir. 2007) (holding that "a transitory, propagating signal" falls outside the scope of patentable subject matter).

74. *See generally* THEODORE ARABATZIS, REPRESENTING ELECTRONS: A BIOGRAPHICAL APPROACH TO THEORETICAL ENTITIES (2006) (providing a history of theoretical representations of the electron as an unobservable entity).

actually exist.[75] A claimed invention that purports to use certain effects of electrons can therefore be patentable. So long as scientists understand how to produce and harness those effects in practice, there is an adequate causal account of the invention's utility. As the Supreme Court declared in *Diamond v. Diehr*, "It is for the discovery or invention of some practical method or means of producing a beneficial result or effect, that a patent is granted."[76]

What allows the theory of electrons to incorporate such a "practical method or means" is our general agreement, for all practical purposes, on what the causal properties of electrons are. In fact, the causal properties of electrons are so fundamental to our modern worldview that they are regarded by many philosophers as metaphysically necessary laws of nature.[77] On this view, all accounts of causal interactions involving electrons are grounded in the essential properties of electrons.

Not all notions of causation are so firmly grounded. Increases in gas prices cause reductions in driving, education causes reductions in poverty and crime, and the assassination of Archduke Franz Ferdinand is regarded as the immediate cause of World War I. But few observers would suggest these causal relationships are of the kind that could give rise to patentable inventions.

In the spirit of a friendly amendment to Lemley's proposal, I propose the following "concrete causation" standard for supporting "structure" under § 112(f):

For a disclosed element to be deemed structural, its involvement in supporting a recited function must be amenable to explanation by a single causal account specifying the resources brought into play (even though such an account need not be known to or submitted by the patent applicant).[78]

In *Diehr*'s terms, I believe this grounding in real-world resource considerations is what should be required of any "practical method or means" of achieving such an effect.[79] Here I use "resources" broadly to refer to any manipulable quantities that have a well-defined causal role generally accepted by practitioners, including physical quantities such as mass, energy, charge, and momentum; and real-time computational resources such as CPU cycles, network bandwidth, memory, disk space, and battery life. Causal accounts may employ a similarly broad range of explanatory principles governing the involvement of such resources in causal processes, ranging from the conservation laws of physics to the scheduling

---

75. *See* BRIAN ELLIS, SCIENTIFIC ESSENTIALISM 146 (2001) ("If the world behaves as if things like atoms and electrons exist, then the best explanation of this fact is that they really do exist.").

76. 450 U.S. 175, 183 n.7 (1981) (citing Corning v. Burden, 56 U.S. 252, 268 (1854)).

77. *See* ELLIS, *supra* note 75, at 6 (arguing that an electron must "generate [certain gravitational and electromagnetic] fields in any world in which it might exist, and have precisely the same effects on things of just the same kinds").

78. A patent applicant has no duty to disclose a correct theory of operation. *See* Newman v. Quigg, 877 F.2d 1575, 1581–82 (Fed. Cir. 1989).

79. *See* 450 U.S. at 183 n.7 (citing Corning v. Burden, 56 U.S. 252, 268 (1954)).

disciplines implemented in operating systems.[80] (As my gas price example suggests, I would exclude notions of economic causation from this picture, but that is a discussion for another time.)

V. CONCLUSION

While my proposal is applicable to all technologies, it would have an especially salutary impact on the problematic software field. Many other commentators have highlighted the apparent ability of artfully drafted software patent claims to evade the abstract-ideas exclusion, but have failed to propose a standard of concreteness that the specification of a patentable software invention must meet.[81] In my view, the most problematically abstract computer-related inventions are those whose results are consequences of mathematics, logic, and semantics, and are therefore indifferent as to the computational resources that may be involved in their implementation.

For example, the commodity-hedging methods the Supreme Court found unpatentable in *Bilski v. Kappos* consist of two kinds of steps: "initiating a series of transactions" and "identifying market participants."[82] The "initiating" steps can be correctly implemented through any kind of process capable of being given legal effect, from paper documents, to recorded phone conversations (the primary approach at the time of Bilski's invention),[83] to HTTP requests via the Internet. The "identifying" steps are specified in purely mathematical terms without regard to the computational resources that might be involved in their implementation.[84] The utility of Bilski's claimed methods is therefore not amenable to one resource-specific causal account, but to many: Bilski's methods perform their hedging functions whether the market participants' option values are calculated on my office desktop PC or on the London Science Museum's Difference Engine, and whether their transactions are completed via telephone or website. A patent examiner could simply cite such an observation in rejecting Bilski's claims as directed to unpatentable subject matter.

My proposed standard of patentable concreteness would be consistent with Supreme Court precedent[85] and could be introduced through either decisional law or legislation. Its universal applicability conforms to our treaty obligations to make patents available without discrimination as to the field of technology, which bodes well for its prospects as an interna-

---

80. *Cf.* PHIL DOWE, PHYSICAL CAUSATION 89 (2000) ("[I]t is the possession of a conserved quantity . . . that makes a process a causal process.").

81. *See supra* Part I.

82. 130 S. Ct. 3218, 3232–33 (2010).

83. *See* FLETCHER J. STURM, TRADING NATURAL GAS 28–29 (1997) (describing natural gas trading by telephone).

84. *See* U.S. Patent Application No. 08/833,892, at A-5 (filed Apr. 10, 1997), *available at* http://www.uspto.gov/go/com/sol/2007-1130bilski_joint_appendix.pdf.

85. *See supra* text accompanying note 76.

tional norm.[86] Finally, it upholds what I have identified elsewhere as the patent system's deep metaphysical commitment to scientific realism.[87]

By design, my proposal explicitly acknowledges that all of the "useful Arts" confront the common problem of having limited resources (in their most general sense) with which to satisfy limitless human desires.[88] This necessity is, after all, the mother of invention.[89] Conscientious programmers and software engineers have acknowledged as much through their attention to the increasingly diverse set of computational complexity concerns that have long suffused and ennobled the discipline of computer science.[90] The patent system exists for those working to do more with less, not for those seeking to corner the market on such efforts.

86. Agreement on Trade-Related Aspects of Intellectual Property Rights, Marrakesh Agreement Establishing the World Trade Organization, Annex 1C, Legal Instruments—Results of the Uruguay Round, Apr. 15, 1994, 33 I.L.M. 1125, 1869 U.N.T.S. 299, at art. 27 (1994).

87. *See* Chin, *supra* note 68, at 5.

88. *Cf.* Jeffrey L. Dunoff & Joel P. Trachtman, *The Law and Economics of Humanitarian Law Violations in Internal Conflict*, 93 AM. J. INT'L L. 394, 395 (1999) (describing economics as "the study of rational choice under conditions of limited resources").

89. *But see* STACY V. JONES, INVENTIONS NECESSITY IS NOT THE MOTHER OF: PATENTS RIDICULOUS AND SUBLIME (1975) (presenting examples of apparently unnecessary patented inventions).

90. *See* SANJEEV ARORA & BOAZ BARAK, COMPUTATIONAL COMPLEXITY (2009) (surveying diverse results in computational complexity theory).