3 Va. J.L. & Tech. 9 (Fall 1998) <http://vjolt.student.virginia.edu>
1522-1687 / © 1998 Virginia Journal of Law and Technology Association

VIRGINIA JOURNAL of LAW and TECHNOLOGY

UNIVERSITY OF VIRGINIA	FALL 1998	3 VA. J.L. & TECH. 9

Patenting Floppy Disks, or How the Federal Circuit's Acquiescence has Filled the Void Left by Legislative Inaction

by Shawn McDonald[*]

I. Introduction

1. Allowing article of manufacture claims to computer instruction (software) as embodied in a computer-readable memory device is a practical method for providing easily enforceable patent protection for the innovation embodied in software. Such claims offer significant enforcement advantages as compared to the process and machine embodiment software claims previously deemed eligible for patent protection. However, allowing patent protection for such claims creates numerous doctrinal inconsistencies when compared to the patent protection afforded inventions claimed in terms of true physical limitations.

2. The fundamental cause of the unpredictability and inconsistency, which might best be described as 'doctrinal chaos,'[1] of recent decisions by the Court of Appeals for the Federal Circuit (CAFC) regarding patent protection for software inventions is simply that the current patent law was not designed to provide such protection. The current patent statute is based upon the mechanical innovation paradigm of the Industrial Revolution rather than the algorithmic innovation paradigm of the current Information Revolution.[2] It is well suited to the protection of tangible machines and industrial processes for converting one substance to another, which represented the great bulk of innovation in 1952.[3] It has also been applied with reasonable success to protect machines and processes for the conversion of electronic signals of one kind into another.

3. However, it has failed to adequately protect systems for processing one kind of data into another kind of data where the physical structure used for implementation, and the symbolic meaning of the data transformed are irrelevant to the invention.[4] The innovation in such cases resides in the idea of how to perform the transformation from one form of symbolic data to another. The current patent statute was designed to protect ideas indirectly by allowing claims to the tangible manifestations of those ideas. In bending the patent law to provide protection for intangible algorithm inventions, lawyers and judges are arguably attempting to allow the direct claiming of ideas.

4. There are compelling arguments both for and against such bending of the patent statute to provide patent protection on algorithm inventions independent from physical and structural claim limitations. The policy arguments on each side are so compelling that the courts have vacillated unpredictably, trying to make policy decisions that are best reserved for the legislature.[5]

II. Non-Patent Forms of Protection for Software Inventions

5. Unlike the mechanical arts in which merely using a product is often sufficient to understand its novelty, computer software functions as a ‘black box’ to transform data into useful information. In fact, an important goal in developing most software products is to insulate the user from the program’s complicated algorithmic machinations through the use of a user-friendly interface. In this way, the user never learns, or even notices, the algorithmic elements used to obtain the desired results.[6] Therefore, even the widespread distribution of software products does not increase public knowledge of software algorithms.[7]

6. Historically, software developers shunned patent protection because of the ineligibility under § 101 of patent claims directed solely to software. A fundamental tenet of the patent system is that protection is provided only for that which the inventor discloses to the public. Because of the unique aspects of software discussed above, developers were accustomed to maintaining their algorithms and software code as closely guarded secrets.[8]

7. Absent patent protection for the software’s underlying idea or concept, developers have relied on the copyright laws to protect the actual expression of the algorithm in the form of the software code itself. No public disclosure is required for copyright protection, but the exclusive rights under copyright are significantly restricted compared to patent rights. A patent gives its holder the exclusive right to make, use, or sell the claimed invention. This not only prohibits others from copying the invention, but it also makes independent discovery an act of infringement. On the other hand, copyright only prohibits copying of the invention; independent creation is not an infringement of the copyright. In this way, the properties of copyright protection further reinforced the incentives for developers to maintain the secrecy of their software code.

8. Secrecy and the copyright laws are ill-suited to protecting software inventions. Secrecy allows the individual software developer to protect the value of her work, but this secrecy creates barriers to innovation and progress within the software industry. Copyright law is designed to protect the individual expression of an idea rather than the idea itself. While providing protection against slavish copying of the program code, it provides no protection for the most valuable aspect of software innovation: the concepts and processes used to implement the program’s novel algorithms, which may be reverse-engineered with impunity under copyright.[9]

9. Denial of patent protection for software has had two important results. First, in an effort to provide protection for the enormous capital investment represented in software products, courts have often attempted to extend copyright protection beyond the literal expression of the software code by

bending copyright law to the point of breaking.[10] Second, because secrecy has remained the strongest form of protection for software there has been little cooperation or sharing of ideas amongst software developers. For this reason, progress in software technology has been woefully slow in comparison to computer hardware and other areas of technology allowed patent protection.

10. There are compelling reasons for providing patent protection to software inventions. However, the current patent statute was not designed for the protection of intangible, algorithmic inventions and the case law indicates the need for legislative guidance.

III. The Birth of the Federal Circuit

11. The current patent statute has remained largely unchanged since its adoption in 1952. Numerous amendments have been enacted, e.g., changing the period of patent protection from seventeen years after issuance to the current twenty years after filing, but the substantive features of the statute remain essentially as they were upon enactment in 1952.

12. The primary purpose of the 1952 Statute was to strengthen patent protection by increasing the predictability of patent infringement and validity litigation. This goal was not realized. In fact, there developed such marked differences between the circuits on the meaning of the statute that forum shopping became notoriously rampant by the 1970s.[11] For example, the Eighth Circuit held every single patent that came before it during the period from 1950 to 1970 invalid.[12] Facing this virtual collapse of the patent system, Congress decided that rather than attempting to once again enact a more specific and detailed patent statute, it would empower a specialized judicial body to 'clean-up' the patent law. It therefore created the CAFC, and established its exclusive jurisdiction over all appeals of Federal District Court decisions involving patents.

13. The message sent by both the legislative and executive branches in creating the CAFC was a clear and adamant request for judicial activism to take the place of legislative debate and resolution of patent law doctrine.[13] The CAFC has done an amiable job increasing the consistency and utility of the patent law, as shown by the increased vigor with which owners of intellectual property have sought patent protection. However, Congress gave the CAFC no new tools in the form of improved patent laws, instead forcing the court to bend the existing patent statute to provide protection for types of innovation unanticipated by the crafters of the 1952 patent statute.[14]

14. The explicit goal of both the Legislative and Executive branches in creating the CAFC was to clarify application of the patent law. This clarification would eliminate the need to address the patent crisis legislatively. A perhaps cynical interpretation of the CAFC's jurisprudence is that in response to pressure from anxious owners of software intellectual property, and in order to insure its continued existence, the CAFC has construed the existing patent statute in an increasingly broad and novel fashion, thereby relieving Congress of the burden of enacting new patent legislation to provide protection for novel forms of innovation.

15. This 'stretching' of the patent law has largely been accomplished through the use of legal fictions

to characterize inventions directed to the novel application of an algorithm in ways that avoid the judicially created *per se* rules barring patent eligibility under § 101.[15] The current position of the CAFC, foreshadowed by Judge Rader in his *Arrhythmia* concurrence,[16] is that when Congress stated that "anything under the sun that is made by man" should be eligible for patentability, they meant it, and therefore judicially created exceptions to patent eligibility should not be imputed to § 101.[17]

IV. § 101 Bars to Patent Eligibility as *Per se* Rules

16. Probably the earliest case in which the Supreme Court was called upon to restrict the breadth of over-reaching claims to an algorithm was *O'Reilly v. Morse*.[18] Morse, in his now infamous claim 8, sought exclusive rights for the use of "electromagnetism, however developed, for marking or printing intelligible characters . . . at any distances."[19] Morse’s disclosure in support of this claim consisted solely of a method and apparatus for amplifying electromagnetic signals at distinct intervals along the transmission path to counteract resistive loss of signal strength and noise intrusion.[20]

17. The Supreme Court disallowed this claim because it was so sweeping in scope as to read upon all future forms of electromagnetic transmission of text, and Morse’s patent application had only disclosed his very basic method, telegraphy.[21] For example, Morse did not disclose electromagnetic facsimile transmission. Yet, because a fax machine electro-magnetically transmits text over a distance it would infringe Morse's claim 8.[22] To hold Morse’s claim 8 valid would have given him a far greater exclusive right than was properly due. In rejecting claim 8, the Court limited Morse’s patent protection to the specific apparatus and method that his patent disclosure placed in the public domain.[23] This enablement requirement is now codified in the patent statute.[24]

18. An alternative explanation for the Court’s invalidation of claim 8 is that the generalized use of electromagnetism cannot be patented because it is a natural phenomenon.[25] Electro-magnetism had been part of the public domain long before Morse’s work, in fact since time began. Morse merely invented a novel way to harness the phenomenon of electromagnetism for telecommunication.

19. Perhaps these two justifications for rejecting Morse’s sweeping claim 8 are rooted in a common theory of patent ineligibility, i.e., granting patent rights for overly broad claim language must be avoided because it removes more from the public domain than the inventor added through disclosure.[26] Evaluating the scope of the patent applicant’s claim in relation to the application’s disclosure for compliance with § 112 may be thought of as a multi-factor ‘rule of reason’ analysis. Conversely, dismissing the claim out of hand as violative of § 101 based solely upon the subject matter of the claim, without reference to the disclosure, may be thought of as applying a *per se* bar to patentability.

20. Like all *per se* rules, the judicially created patent ineligibility doctrine under § 101, such as the exclusions for business methods and algorithms, increases predictability and reduces the adjudication costs of patentability determinations. However, use of the § 101 *per se* bars to patent eligibility increases the likelihood of error in any individual patentability determination because their application is mechanical and fails to consider the merits of each case.[27]

V. Algorithm Precedents: The Supreme Court

A. Gottschalk v. Benson

21. The first Supreme Court decision addressing the patent eligibility of computer-implemented algorithms was *Gottschalk v. Benson*.[28] Benson sought patent protection on a process for the conversion of binary-coded decimals into pure binary numbers, which may be directly manipulated by a digital computer. The only physical limitation on the scope of the claims, which was absent from one of the two claims before the Court, was that a 'shift register' (a generic article of computer memory hardware) be utilized for implementation of the claimed process.[29] In rejecting Benson's claims as ineligible for patent protection under § 101, the Court concluded that the scope of the claims was so sweeping that the underlying concept or idea that was the foundation of Benson's process would be removed from the public domain.

22. Justice Douglas summarized the Court's view of Benson's process claims in his famous 'nutshell':

> It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting BCD numerals to pure binary numerals were patented in this case. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself. . . . If these programs are to patentable, considerable problems are raised which only committees of Congress can manage, for broad powers of investigation are needed, including hearings which canvass the wide variety of views which those operating in this field entertain.[30]

23. Douglas reasoned that the claimed invention was ineligible for patent protection because it could only be practically applied using digital computer technology. Would the Court have found the claims eligible for patent protection if they could have been implemented through several different technologies, or if Benson had limited his claims to a specific use?[31]

24. In objecting to the scope of Benson's claims under § 101, the Court fashioned a *per se* rule of ineligibility for algorithm claims whose effect could be avoided only through the inclusion of physical limitations in the claims, i.e., claims to the implementing hardware.[32] The *Benson* rule for eligibility under § 101 stands on questionable doctrinal ground. The subject matter of a claim, rather than its scope, is the sole ambit of § 101 eligibility analysis.[33] The Court's objections to

the scope of Benson's claims may have been more properly made under the 'rule of reason' requirements of disclosure and enablement under § 112.[34]

25. The *Benson* opinion introduced some of the themes which would recur throughout algorithm case law. First, patents should not be granted for inventions comprising the "basic tools of scientific and technological work" because granting exclusive rights over their use would retard progress.[35] Second, to what extent must process claims include limitations involving the "transformation and reduction of an article 'to a different state or thing'?"[36] Finally, the Court called for legislative guidance on the issue of patent eligibility for algorithms.[37] This call, still unanswered, is a mantra found throughout the subsequent Supreme Court and CAFC decisions regarding the patenting of computer innovation.

B. Parker v. Flook

26. The Supreme Court's next algorithm opinion, *Parker v. Flook,*[38] provided what would prove to be a short-lived framework for consideration of the patent eligibility questions originally raised by the *Benson* opinion. The patent applicant in *Flook* had attempted to avoid running afoul of the Court's requirements as stated in *Benson* by expressly limiting the claimed field of use to catalytic conversion of hydrocarbons and by including post-solution activity limitations in the claim, in this case the activation of a warning buzzer if the claimed algorithm's resultant value exceeded a specified alarm limit.[39]

27. In holding Flook's claims ineligible for patent protection, Justice Stevens' opinion for the majority announced that the novelty of the claimed algorithm is immaterial to the determination of patent eligibility under § 101.[40] Under Stevens' eligibility test the algorithm was treated as though it were prior art; similarly, in *Beauregard* the PTO had sought to avoid consideration of the informational content of the claimed computer disk in its § 103 obviousness determination by classifying it as prior art under the printed matter doctrine.

28. After removing the algorithm element of the claim from the § 101 eligibility determination, Stevens concluded that, "[R]espondent's process is unpatentable under § 101, not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention."[41] Thus, it was not sufficient that Flook had limited his claims to a specific field of use and included post-solution physical activity; those additional claim limitations would also have to satisfy novelty and nonobviousness requirements for the claim to be eligible for patent protection.[42] This created an unworkable threshold which functioned as a *per se* bar to patent eligibility for algorithms claimed as processes. Why would an inventor of novel hardware seek to limit the exclusive rights to such hardware by including software claim limitations that would be afforded no patentable weight by the PTO, but would be construed as limitations to infringement by the courts?[43] Therefore, the net effect of *Flook* was to reinforce the anti-software, pro-hardware claiming bias for computer inventions.[44]

C. Diamond v. Diehr

29. The Supreme Court's most recent opinion addressing the patent eligibility of software inventions, *Diamond v. Diehr*,[45] expressed a view fundamentally different from its *Benson* and *Flook* decisions. With *Diehr* the Court began a process of invalidating the bars to § 101 eligibility that it had announced in *Flook*.[46] This weakening of *Flook* continued unabated in the CAFC's subsequent decisions.

30. The claimed invention in *Diehr* was a computerized algorithmic process for use in controlling rubber molding equipment. The Court, in a 5-4 split, overturned the PTO's rejection of ineligibility under § 101, holding the process claims eligible as being directed to more than just the prior art thermodynamic equation used to control the process.[47] In arriving at this conclusion, the Court rejected its approach in *Flook* of dissecting out the algorithm portion of the claim when determining eligibility under § 101.[48] The Court did, however, reaffirm *Benson* and *Flook* as authority for the § 101 bar to eligibility for mathematical formulae standing alone.[49]

31. In *Diehr*, the Court placed a great deal of emphasis on the physical steps included in the claims as evidence that more than simply an algorithm was claimed, distinguishing *Flook* on this ground. It is unclear why detailed disclosure of the physical steps included in the claimed process should be determinative, especially if control of a rubber molding process is the only practical application of the equation.[50] If that were the case, the patent in *Diehr* would completely preempt practical use of the equation, just as the Court had feared in *Benson*.[51]

32. Since *Diehr*, the Supreme Court has left adjudication of algorithm patents to the CAFC. Unfortunately, the Court's decisions do not provide consistent precedent, and the CAFC has struggled to arrive at a predictable eligibility rule for claims involving algorithms.

VI. Algorithm Precedents: 1994--The Federal Circuit's Year of the Algorithm

33. In 1994 the CAFC handed down five key decisions regarding software patents: *In re Schrader*,[52] *In re Alappat*,[53] *In re Warmerdam*,[54] *In re Lowry*,[55] and *In re Trovato*[56]. Analysis of these decisions provides the most reliable framework for predicting the CAFC's evaluation of *Beauregard*-type article of manufacture claims under §§ 102, 103, and 112. Unfortunately, only *Alappat* was an *en banc* decision. Several commentators have argued that these five decisions are inconsistent and do not indicate any significant change in the CAFC's consideration of software patents.[57]

A. In re Schrader

34. In *Schrader*, the first of the five decisions to be handed down, a 2-1 split panel affirmed the PTO’s rejection of Schrader’s claims as being drawn to ineligible subject matter in violation of § 101. Judge Plager, writing also for Judge Mayer, ruled that the applicant’s invention as claimed was unpatentable because it was solely directed to an algorithm, with no structural limitations as required by *Flook* and *Diehr* to bring it within § 101’s eligibility requirements. The claimed invention was a method useful for auctioning mixed lots of related items in real time.[58] The CAFC based its finding of ineligibility on the fact that there were no ‘structural’ limitations on the claimed use of the algorithm, i.e., the patent claims were solely directed to a process of bid data analysis with no physical limitations.[59] The panel ruled that such a process would only be patentable if claimed in combination with a transformation of either physical elements or data that is directly representative of physical elements.[60]

35. Judge Plager applied the two-step Freeman-Walter-Abele (FWA) test,[61] the first step of which is to determine whether a mathematical algorithm is recited directly or indirectly in the claim. Judge Plager found that Schrader’s claimed process was implicitly directed to a mathematical algorithm, as evidenced by the claim language "assembling a completion."[62] The second step of the FWA test is to determine whether the claimed invention is no more than the algorithm itself.[63] Schrader’s patent specification disclosed his best-mode implementation, explaining how auction participants would observe the real-time status of their bids on large video display units and submit new bids to a central location via telecommunication, where the claimed process for bid analysis would be implemented with the updated results instantaneously displayed on the before-mentioned video display units.[64] However, Schrader’s claims contained no mention of the video display units, or any other physical effect or result,[65] and there was no basis to read such limitations from the specification into the claims.[66]

36. The fundamental flaw of the FWA test is that practically every process may be expressed mathematically. That is the essence of the utility of mathematical analysis. Therefore, since the symbolic language of mathematics may be used to define practically all processes, it is also true that a process patent claim embodied in symbolic language is implicitly directed to a mathematical algorithm. Once this is admitted, the FWA test rests solely on its second prong, and therefore § 101 eligibility obtains only if the claim includes significant physical limitations.

37. This collapsing of the FWA test into a simple search for physical claim limitations, a la *Diehr*, results in no improvement in the predictability or accuracy of § 101 eligibility determinations. Indeed, in applying the second prong of the FWA test to Schrader’s claims Judge Plager found no physical limitations and, therefore, no § 101 eligibility. However, a CAFC panel comprising Judges Newman, Lourie, and Rader had ruled previously that claims to a computerized process for analyzing data representing cardiac impulses were eligible under § 101.[67] In attempting to distinguish *Arrhythmia* and *Abele* on their facts, Judge Plager found that the data in those cases were "representative of or constituting physical activity or objects."[68] Rather than holding that the bid data in Schrader’s claims were representative of the physical activity of bidding in an auction, Judge Plager found that "Schrader’s claims, except for incidental changes to a ‘record,’ do

not reflect any transformation or conversion of subject matter representative of or constituting physical activity or objects."[69]

38. A clear rule for determining § 101 eligibility is discernable from the majority decision in *Schrader*; however, the doctrinal justification for this rule is not compelling. *Schrader* established two alternative paths to eligibility for computer inventions under § 101: (1) include physical apparatus for implementing the process in the claim, or (2) establish that the data signals manipulated by the algorithm are representative of physical activity or tangible objects.[70] If the claims do not include such limitations they will be rejected as claiming the algorithm in isolation. The simple, highly predictable nature of this rule, when combined with its questionable utility as a judicially created *per se* limitation on § 101 eligibility, led one commentator to label it a "bright zig-zag rule."[71] Would Schrader's claims have been found eligible under § 101 if he had included superfluous physical limitations in the claims such as prior art video display units? That is the type of claiming trickery that Judge Nies protested in *Trovato*.[72]

39. Judge Newman's dissent in *Schrader* is more compelling for a number of reasons. First, in compliance with the Supreme Court's nullification of judicially created exceptions to § 101 eligibility in *Diehr*, she would find eligibility under § 101 but remand the case to the PTO for further evaluation of the claimed invention to determine compliance with the statutory requirements for novelty, nonobviousness, disclosure, and enablement as codified in §§ 102, 103, and 112.[73] Judge Newman interpreted the second prong of the FWA test, shown above to be the only prong of practical importance, much less restrictively than the majority, requiring only that the algorithm be claimed as part of a useful process.[74] Judge Newman's interpretation of the FWA test is consistent with both Supreme Court precedent implicitly limiting § 101 to its statutory language (thereby invalidating judicially created exceptions to eligibility)[75] and earlier CAFC decisions regarding § 101.[76]

40. In its alternative rejection of Schrader's claims based on the judicially created § 101 exception for methods of doing business, the PTO found case precedent on the business methods exception contradictory and sought guidance from the CAFC. In honoring this request, Judge Newman called for the retirement of the business method exception to § 101 because it was "error-prone, redundant, and obsolete."[77] Rather than applying a judicially created *per se* rule of questionable legality and prudence, Judge Newman called for patentability determination under the 'rule of reason' analysis codified in §§ 102, 103, and 112 of the patent statute.[78]

B. In re Alappat

41. As the only *in banc* CAFC decision on eligibility under § 101, *In re Alappat* stands as the landmark decision on patent eligibility of computer inventions. In *Alappat* the CAFC also provided guidance on the proper procedure for examination by the PTO of claims written in means-plus-function form per paragraph six of § 112, which permits an applicant to claim an element of an invention in functional language, such as "means for doing X."

1. Means-plus-function Claims

42. The scope of a means-plus-function claim includes all structures specifically described in the patent specification as a means for accomplishing that function, and their equivalents. This is important when the inventive step lies elsewhere in the claimed process or article, and the use of specific structural descriptions in the claim would unnecessarily restrict the claim's scope, thereby unnecessarily limiting the patent holder's exclusive rights. Means-plus-function language is especially useful in claiming computer inventions because any given element of the claimed process may be accomplished using an almost endless array of software/hardware algorithm choices, even though they each result in an equivalent effect.[79] If a patent holder's exclusive rights were restricted to specific claim language it would be simple for would-be infringers to design around the elements specifically claimed or disclosed in the patent, thereby stealing the value of the patent holder's invention without infringing the patent.[80] Unfortunately, proper determination of the scope of equivalents for means-plus-function language is difficult, and prior to its *Hilton Davis* decision in 1995,[81] the CAFC had not formed an internal consensus on the appropriate legal standard.[82]

43. In attempting to avoid the problems associated with determining which prior art structures were equivalent to the structures specifically disclosed in a patent application, the PTO adopted a rule that means-plus-function claims would be interpreted as reading on any reasonable means whatsoever for accomplishing the function. This reduced the complexity and unpredictability of examining means-plus-function claims, but it also meant that many such claims were unfairly disallowed as obvious under § 103 in light of the prior art because the PTO was interpreting the scope of the claim more broadly than it would later be interpreted by the courts in infringement litigation.[83]

44. In *Alappat*, the CAFC directed the PTO to interpret the scope of means-plus-function claims as limited to the equivalents of the element as described in the patent specification.[84] This requirement allows patent applicants to craft means-plus-function claims in combination with their specification disclosure that will be allowed by the PTO and which will provide a commercially significant scope of exclusivity, i.e., exclusive rights to their invention and its equivalents. This directive in *Alappat* paved the way for *Beauregard*'s means-plus-function claims to computer instruction as embodied in a computer-readable memory device.

2. § 101 Eligibility Requirements

45. While the means-plus-function portion of the majority decision in *Alappat* is clear, the holding regarding § 101's eligibility requirements is less so. Six judges, Rich, Newman, Lourie, Michel, Plager, and Rader, agreed on a majority opinion, which was written by Judge Rich. Alappat's claim 15,[85] the only one on appeal, was held eligible under § 101, and found to encompass a machine that implemented an anti-aliasing algorithm, rather than the algorithm in isolation, i.e.,

the claim complied with § 101 because it did not preempt all use of the algorithm.[86]

46. Alappat's invention, the elements of which were claimed solely in means-plus-function language, is a 'rasterizer' which modifies oscilloscope input data in order to improve the apparent resolution of the output data when observed on a visual display means (such as a CRT). In essence, Alappat's invention is an improvement in an oscilloscope similar to a television having a clearer picture.[87] Image enhancement of this type for CRT displays was well known in the prior art; Alappat's innovation was a novel anti-aliasing algorithm for conveniently calculating the proper data modifications necessary to transform the input data into output data with the improved, anti-aliased appearance.

47. Like all algorithms, Alappat's anti-aliasing algorithm could be implemented using solely hardware components, solely software elements (i.e., microprocessor instructions), or any combination of the two.[88] Likewise, Alappat's means-plus-function language chosen for claim 15 would read on any combination of software and hardware that implemented his anti-aliasing invention. The issue before the CAFC was whether the scope of claim 15 was illegally broad, claiming all use of the algorithm itself rather than just machines that implemented the algorithm.

48. The majority held that claim 15 was not violative of § 101; its scope did not include the algorithm standing alone, but rather was limited to a machine which implemented the algorithm and which contained the elements specifically claimed.[89] If one accepts the majority's interpretation of claim 15, then this holding of eligibility under § 101 is reasonable and consistent with earlier decisions such as *Schrader*.[90]

49. The PTO had also rejected Alappat's claim 15 as unpatentable because it reads on a general purpose digital microprocessor operating software designed to implement the anti-aliasing algorithm.[91] The CAFC reversed this rejection, Judge Rich writing:

> Alappat admits that claim 15 would read on a general purpose computer programmed to carry out the claimed invention, but argues that this alone also does not justify holding claim 15 unpatentable as directed to nonstatutory subject matter. We agree. We have held that such programming creates a new machine, because 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.[92]

50. Implicit in the above passage is the conclusion that upon being programmed to implement the claimed invention, i.e., the algorithm, a microprocessor, standing alone, becomes a 'special purpose computer' which infringes the claim. With this conclusion, the majority in Alappat went beyond all precedent and created the most sweeping patent protection for computer inventions yet endorsed by the CAFC. The passage above indicates that a majority of the CAFC no longer support the requirement that claims to computer inventions include specific apparatus limitations in order to qualify for eligibility under § 101.[93]

51. The dissent from Judges Archer and Nies objected to the sweeping scope of protection attributed to claim 15 by the majority. In finding that a general purpose computer would infringe claim 15 when running software which practiced Alappat's algorithm, despite the fact that the computer had none of the hardware elements included in claim 15, the majority had given Alappat exclusive rights over all methods to practice his algorithm, and therefore patent rights in the algorithm itself.[94] In allowing the means-plus-function language of claim 15 to cover all possible hardware and software combinations used to practice Allapat's algorithm, the majority had given patent rights to a mathematical equation, in violation of all relevant precedent from the Supreme Court, the CCPA, and the CAFC itself.[95]

52. At first blush, the protestations by Judges Archer and Nies seem compelling. However, upon closer examination it becomes clear that the arguments posited by the dissent for denying § 101 eligibility are no more principled than the majority's arguments in favor of eligibility. Chief Judge Archer found Alappat's claim 15 ineligible under § 101 because:

> Alappat admits that each of the circuitry elements of the claimed "rasterizer" is *old*. He says they are merely "form." Thus, they are only a convenient and basic way of electrically representing the mathematical operations to be performed, that is, converting vector data into matrix or raster data. In Alappat's view, it is the new mathematic operation that is the "substance" of the claimed invention or discovery. Claim 15 as a whole thus claims *old* circuitry elements in an arrangement defined by a mathematical operation, which only performs the very mathematical operation that defines it. Rather than claiming the mathematics itself, which of course Alappat cannot do, Alappat claims the mathematically defined structure. But as a whole, there is no "application" apart from the mathematical operation that is asserted to be the invention or discovery. What is going on here is a charade.[96]

53. The only 'charade' here is Chief Judge Archer's attempt to justify an eligibility rejection under the *per se* framework of § 101 by showing a lack of novelty in the invention as claimed, which is only appropriate under the 'rule of reason' novelty determination of § 102. Not only is the novelty attack inappropriate when the decision under appellate review is a § 101 rejection, but Chief Judge Archer missed his target on the merits as well. Patentability is never negated by a lack of novel *components*. The claimed invention need only be novel and nonobvious when considered as a whole;[97] after all, only God creates from novel components.[98] Viewed as such, Alappat's claimed *combination* of prior art circuit elements is certainly a novel article of manufacture, and therefore his claim is eligible under § 101, and the determination of patentability should proceed to consideration of the requirements in §§ 102, 103, and 112.

54. The fundamental problem in *Alappat* was the inability of the current patent statute to allow both the principled determination of which computer inventions are worthy of patent protection, and the provision of adequate protection against infringement under a workable doctrine of equivalents. Chief Judge Archer's dissent was prophetic in its prediction of the difficulties that lie beyond the § 101 patent eligibility questions addressed in *Alappat*:

> Because the patent law does not examine abstract mathematics, if the "rasterizer" is held to be within § 101, there can be no meaningful examination for compliance with § 103, and other sections of the patent statute become inapplicable. The practical result is that there is patentability so long as the mathematics is "new." . . . Alappat cannot have it both ways. If a programmed general purpose digital computer is not statutory subject matter, then a claim cannot be drawn to that subject matter whether outright or by application of equivalents under 35 U.S.C. § 112, ¶ 6. Paragraph 6 of § 112 is not a magical way to expand patent protection into nonstatutory subject matter.
>
> As to equivalency, finding equivalency in a programmed general purpose computer proves the nonstatutory nature of Alappat's purported invention or discovery. Alappat argues that the electrical circuitry of the "rasterizer" is equivalent to a programmed general purpose computer because "powerful, inexpensive microprocessors" are equivalent to "discrete digital components, such as AND, OR, NAND, etc., gates, registers, latches, and the like" are equivalent to "analog components, such as transistors, operational amplifiers, and resistors." They are all equivalents, in Alappat's view, because they all may achieve the same effect: performing the particular mathematics that is the claimed rasterizer.[99]

55. Both the majority and dissenting opinions in *Alappat* are actually comprised of two distinct determinations. The majority is almost certainly correct in holding Alappat's claim 15 eligible under § 101, however in creating a spectrum of equivalents under § 112 ¶ 6, that is so broad as to read on a general purpose microprocessor programmed to implement Allapat's algorithm, the majority has impermissibly expanded the scope of protection so far that claimed physical limitations are irrelevant. In finding equivalency for a claimed computer invention after *Alappat*, only the result matters, the function and way are immaterial.[100]

56. As identified by Chief Judge Archer, the majority decision in *Alappat* offers no guidance on the proper application of §§ 102, 103, and 112 with regard to computer inventions. Unfortunately, other than Judge Rader's § 103 obviousness analysis of the claims in *Lowry*, there has still been no guidance from the CAFC on these issues, which are considerably more difficult than the eligibility question under § 101.

C. In re Warmerdam

57. The other three 1994 algorithm cases, *In re Warmerdam,*[101] *In re Lowry,*[102] and *In re Trovato*[103] are perhaps only useful as evidence that the decision in any algorithm case before the CAFC is more a function of the composition of the three judge panel hearing that case than the facts at issue.[104] Even two of the judges concurring in the majority *Alappat* decision came to arguably inconsistent conclusions in *Lowry* and *Warmerdam*.

58. Judge Plager, writing a unanimous decision for Judges Lourie, Clevenger, and himself in *Warmerdam*, held the applicant's claims to a data structure that was useful in modeling a three

dimensional environment ineligible for patent protection under § 101.[105] Likewise, the process claims for the method of creating such data structures were also held ineligible under § 101.[106] After searching for physical claim limitations on which to base § 101 eligibility, as he did in *Schrader*,[107] and finding none, Judge Plager ruled that, "As a whole, the claim involves no more than the manipulation of abstract ideas."[108] This decision is completely consistent with Judge Plager's opinion in *Schrader*, and it appears that the claims in *Warmerdam* would have been deemed eligible under § 101 had they included even basic physical limitations. Ironically, the scope-limiting effects of such limitations had arguably been negated by the *Alappat* decision.

D. In re Lowry

59. In *Lowry*, Judge Rader, writing for a unanimous panel including Judges Rich and Skelton, overruled the PTO's rejection of Lowry's claims as non-novel under § 102 and obvious under § 103. Judge Rader held that Lowry's claims to a memory comprising a novel and nonobvious hierarchical data structure satisfied §§ 101, 102, and 103. Similar to the *Beauregard* rejection, the PTO had refused to afford patentable weight to the data structure in the nonobviousness analysis under § 103, likening the data structure to printed matter. Judge Rader's opinion strongly overturned this application of the printed matter doctrine to computer memory structure, holding this exception to patent eligibility applicable, if at all, only to information intended to be read by humans.[109] Based upon *Lowry* the PTO withdrew its printed matter rejection in *Beauregard*, and moved for dismissal of Beauregard's CAFC appeal.[110]

60. As Judge Plager had done in *Warmerdam*, Judge Rader looked for physical limitations in Lowry's claims. However, unlike Judge Plager he found them: "[m]ore than mere abstraction, the data structures are specific electrical or magnetic structural elements in a memory."[111] This determination that configuration of a computer memory results in actual physical structure is completely consistent with the majority opinion in *Alappat*, where Judge Rich wrote that the configuration of a general purpose computer with specific software creates a new, special-purpose machine.[112]

61. The *Lowry* case was the CAFC's first opportunity to provide guidance on the proper application of § 103 to computer inventions. While it is apparent that Judge Rader and his clerks had a strong command of the subject technology, the conclusion of nonobviousness rests heavily on the absence from the prior art reference of the specific, pyramidal hierarchy claimed by Lowry. However, Lowry's hierarchical structure could easily have been created using the prior art system for constructing data structures. The patentable nonobvious utility of Lowry's claimed data structure resides in the increased efficiency, relative to prior art data structures, with which the structure may be utilized by the implementing microprocessor. The rub is that while there are distinct structural differences between Lowry's claimed data structure and the prior art sufficient to satisfy the novelty requirements of § 102, it is unclear whether these differences are indeed nonobvious. This is, of course, a judgment call, but Judge Rader's opinion provides little guidance for future § 103 determinations of this kind because there is only a single prior art reference with

which Lowry's claims may be compared.

62. The *Warmerdam* and *Lowry* decisions were handed down within two weeks of one another, and in the shadow of the *en banc* consideration of similar issues in *Alappat*. Yet, given the differences between the *Warmerdam* and *Lowry* decisions, it is at least arguable that Judge Rader's panel would have found both inventions eligible under § 101, while Judge Plager's panel would have found neither eligible.

E. In re Trovato

63. The last CAFC algorithm decision in 1994, *In re Trovato,*[113] involved claims to a data structure useful in the calculation of minimum distances between objects in three dimensional space. These claims were very similar to those in both *Warmerdam* and *Lowry*. Two types of claims were at issue: (1) process claims for constructing the data structure in order to model a three dimensional environment (similar to the claimed process in *Warmerdam*), and (2) apparatus claims directed to a machine used to implement such a process.[114]

64. The panel in *Trovato*, consisting of Judge Nies, Judge Michel, and Judge Schall, unanimously upheld the PTO's rejection of all claims as directed to nonstatutory subject matter and therefore ineligible for patent protection per § 101. Judge Nies, writing for the panel concluded that the specifications "provide no grasp of underlying physical process," and the process claims were directed solely to "the process of performing a numerical calculation."[115] Judge Nies dismissed the apparatus claims as a claim drafting subterfuge, "The use of an apparatus claim format in this fashion is precisely the sort of guise recognized in *Alappat* and the cases cited therein."[116] *Alappat* was cited in support of the above conclusions, despite the fact that the majority's opinion in *Alappat* recognized the complete interchangeability of modern hardware and software components, and addressed this interchangeability dilemma by ruling that the means-plus-function claim in *Alappat* would read equally on software and hardware implementations of the claimed invention.

65. The *Trovato* panel rejected the apparatus claims because, "Trovato does not claim to have invented a new kind of computer which the recited mathematical algorithm controls."[117] This clearly contradicts the majority's conclusion in *Alappat*, "We have held that such programming creates a new machine, because 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."[118] The *Trovato* decision was also inconsistent with the decision in *Lowry*, but these inconsistencies were not even mentioned in the *Trovato* opinion.

66. The *Trovato* decision was roundly criticized by commentators,[119] and the CAFC acted *in banc* in 1995 to withdraw the panel decision, not on the merits but rather to remand it for reconsideration in light of the *Alappat* decision and forthcoming PTO examination guidelines.[120]

67. The fact that a decision of such questionable authority as *Trovato* was supported by three of the eleven judges on the CAFC is indicative of both the profound personal differences of opinion among CAFC judges, and the lack of precedent sufficient to curb their willingness to decide cases based upon those personal opinions.

F. Significance of the 1994 Decisions

68. In considering the five algorithm cases handed down by the CAFC in 1994, it appears that the law of patents as applied to computer inventions is in a state of disarray similar to that of the patent law generally prior to the creation of the Federal Circuit.[121] Each CAFC judge has tried amiably to address policy issues that are only effectively handled by the Congress. What has ensued, even after the *in banc* consideration of *Alappat*, is a Court of Appeals where the identity of the three judges deciding the appeal has an equal, or greater, effect on the decision than do the facts of the case. This is similar to the situation prior to the creation of the CAFC when the identity of the Circuit hearing the case was a more reliable predictor of whether the patent would be upheld or invalidated than the facts of the case. Only now, appellants cannot 'forum shop'; they are stuck with whichever three judge panel their appeal draws.

69. It is extremely difficult to discern a consistent 'rule' for determining eligibility under § 101 from the five 1994 decisions, in fact it seems most likely that there is none. However, one commentator has proposed that upon examination of the mode of analysis employed in the 1994 decisions, excepting *Trovato*, a clear pattern emerges showing the adoption of a new two-step test for eligibility under § 101.[122] Laurenson proposes that a majority of six CAFC judges (Rich, Newman, Plager, Lourie, Clevenger, and Rader) have adopted this two-step test which results in eligibility under § 101 provided that: (1) the claim describes the software invention as embodying, representing, or being intimately associated with one of the four basic § 101 categories of eligible subject matter, i.e., a process, machine, article of manufacture, or composition of matter, and (2) the claim encompasses more than an abstract idea or mathematical formula.[123]

70. Laurenson is perceiving doctrinal unity where none exists. To propose that Judges Rader and Rich are in agreement with, and applying the same eligibility test as, Judges Plager, Lourie, and Clevenger is to ignore the fundamental differences between their holdings in *Lowry* and *Warmerdam*. Laurenson is correct in concluding that these six judges look to the same factors when making a § 101 eligibility determination, but he misinterprets this shared analysis as agreement on the eligibility determination itself. As shown by *Lowry*, *Warmerdam*, *Alappat*, and *Arrhythmia*, Judges Rader and Rich are uneasy about denying patent protection under the *per se* rule of § 101, preferring instead to allow a detailed rule of reason analysis of patentability under §§ 102, 103, and 112, while Judges Plager, Lourie, and Clevenger are most concerned with protecting the public domain of abstract ideas and mathematical formulae from over-reaching patent claims, and for which the *per se* rules of § 101 are most effective.

VII. Errors in the Patentability Determination

71. The decision whether to grant patent protection for an innovation may result in two different types of error. Exclusive patent rights may erroneously be allowed for something that is in fact not a patentable invention, i.e., either actual prior art or obvious in view of prior art. This ‘false positive’ patentability determination may be referred to as "Type I" error. The inequitable benefit of Type I error inures solely to an individual, the patent holder, while the inequitable cost of Type I error is borne by society generally. Type I error results in the removal of a discreet item from the public domain which is then placed under an individual’s exclusive control. This is analogous to an individual fencing off a portion of the town commons for her exclusive use.

72. The second type of error occurs when patent protection is erroneously denied for an innovation that is in fact novel and nonobvious. This ‘false negative’ patentability determination may be referred to as "Type II" error. The inequitable benefit of Type II error inures to society as a whole, i.e., all may freely practice the innovation, while the inequitable cost of Type II error is borne solely by the inventor whose innovation has been placed in the public domain. Type II error results in the removal of a discreet item from the inventor’s exclusive control which is then placed in the public domain. This is analogous to the town council taking a portion of an individual’s land to use as a commons without reimbursement.

73. Both types of error are undesirable because they reduce both allocative efficiency and the predictability of allocative determinations. The desire to avoid such errors has resulted in very detailed, fact specific rules for determining patentability. These multi-factor ‘mud’ rules comprise the bulk of the patent statute. While these ‘mud’ rules help to avoid errors in individual patentability determinations, they increase the effort, i.e., cost to society, required in making those determinations.

74. Alternatively, while ‘crystal’ *per se* patentability rules increase the overall error in patentability determinations,[124] they also increase the predictability of such determinations. This is beneficial because the increased predictability/reliability of patent rights reduces transaction costs.[125] If there is 100% certainty that a patent on a novel, nonobvious software invention will be upheld as valid in later infringement litigation, then the value of that patent will be the actual market value of the exclusive right to that innovation. However, if there is only a 60% certainty that the patent will be upheld as valid, then its value may be substantially less.

75. If the decreased certainty of validity was due strictly to judicial unpredictability then the value of all software patents, sticking to the example above, will be discounted by a fixed 40%. However, if the uncertainty was due to the application of a multi-factor validity rule, then accurate valuation of a specific patent would require investigation of the individual circumstances surrounding that patent so that the probability of its being upheld could be determined. Under this regime, the value of a patent would be discounted by its specific likelihood of not being upheld as valid under the multi-factor ‘mud’ rule (plus the cost of acquiring the information necessary to perform such an analysis).

76. This analysis results in an inescapable tension between the desire to insure that individual

patentability determinations are ‘fair,’ and the conflicting desire to reduce transaction costs by insuring predictable valuation of patent rights with minimal information costs. The decision regarding how to address these mutually exclusive goals is of fundamental importance, and in a democratic government should be determined by the legislature.[126]

77. A legislative decision to increase the predictability of patentability determinations, and/or reduce or eliminate *either* Type I or Type II errors[127] would be implemented by amending the patent statute to increase the role of ‘crystal’ *per se* patentability rules such as the § 101 patent eligibility requirements. Similarly, a decision by the legislature to reduce the combined occurrence of Type I and Type II error would be implemented by amending the patent statute to increase the role of multi-factor ‘mud’ patentability rules such as the requirements of §§ 102, 103, and 112 of the current patent statute.

78. In the absence of guidance from the legislature, the judges on the CAFC have allowed their personal opinions regarding the policy choices discussed above to inform their decisions, resulting in the disunity observed in the CAFC’s determinations of patentability for computer inventions.

79. As an example, the framework outlined above may be applied to the decisions in *Warmerdam* and *Lowry*. Judge Plager’s opinions in *Warmerdam* and *Schrader* convey his conviction that claims to abstract ideas and mathematical formulae must be avoided at (almost) all costs.[128] It is a natural choice, then, for Judge Plager to apply a ‘crystal’ *per se* rule against the patentability of claims that may be construed as reading on abstract ideas and formulae. Judge Plager does this through broad interpretation and application of the *per se* § 101 prohibition on claiming algorithms.

80. Alternatively, Judge Rader’s opinions in *Arrhythmia* and *Lowry* convey his preference for basing patentability decisions on the individual merits of each case. Therefore, it is natural for Judge Rader to shun the application of *per se* prohibitions to patentability under § 101, in favor of a multi-factor ‘mud’ patentability analysis under §§ 102, 103, and 112, such as he undertook in *Lowry*.[129]

VIII. In re Beauregard

81. After the CAFC’s five 1994 algorithm decisions discussed *supra*, it is clear that in order to be held eligible under § 101 a claim must include a structural element to insure that it is directed to more than just the algorithm. The specific structure chosen for inclusion in the claim has little legal significance, but it may have a profound effect upon the real-world enforceability of the claim. There can only be infringement when the claimed invention is practiced through the structure included in the claim.

82. In the case of claims such as those at issue in *Alappat* and *Lowry*, only the end-user of the claimed invention could be held liable for infringement because only they practice the claimed invention through use of the claimed structural hardware. This type of protection is of little value to software inventors because chasing down every end-user of infringing software would be much more

costly, and much less rewarding, than pursuing an infringement claim directly against the producers and distributors of software which makes infringement by the end-users possible.[130]

83. *In re Beauregard*[131] was IBM's attempt to impart § 101 eligibility to a claim through structural limitations that would not prevent an infringement suit directed to the producers and distributors of allegedly infringing software. This was done by claiming the invention as an article of manufacture comprising a storage device, e.g., floppy diskette or CD-ROM, encoded with machine-readable software code for implementing a novel and nonobvious algorithm. The software elements of the claim consisted of a series of means-plus-function clauses. Because the PTO had already issued a presumably valid patent to IBM for method claims and system claims including more extensive structural limitations, the only issue in *Beauregard* was whether a floppy disk containing the implementing software code was eligible for patent protection.[132]

84. The PTO rejected the claims in *Beauregard* as both nonstatutory, in violation of § 101, and obvious, in violation of § 103. However, the § 103 rejection was a result of the PTO's application of the printed matter doctrine and, therefore, not a true obviousness determination in view of prior art.

A. Application of the Printed Matter Doctrine in Beauregard: A Cheat by the PTO

85. The printed matter doctrine is a judicially created exception to eligibility under § 101 whereby claim elements which are classified as 'printed matter' are given no weight in the § 101 eligibility determination, and may be disregarded as well in the § 103 obviousness determination. The purpose of the printed matter doctrine, as expressed in the CCPA precedents relied upon by the PTO, is to deny patentability for an article of manufacture, such as a book, the sole novelty of which is the artistic expression contained in the symbolic type.[133] Protection for such artistic expression has been provided by copyright law, and it is reasoned that the exclusionary rights conferred by a patent are too sweeping to be awarded solely on the basis of artistic expression.

86. The PTO rejected the claims in *Beauregard* as nonstatutory under § 101 by classifying the software code contained in the storage medium as printed matter, and then removing it from the § 101 determination through application of the printed matter doctrine. In affirming the § 101 rejection, the PTO Board of Appeals reasoned that the novel algorithm was encoded in a prior art software language, the method of storage (the floppy disk) was also prior art, and therefore the sole novelty was analogous to expression in the language of software code. The Board found no novel relationship between the software code and the substrate (a test developed in the printed matter case law to distinguish between functional and expressive markings) and, therefore, applied the printed matter doctrine to find no patentable invention. Given the above, affirmance of the PTO's § 103 rejection was a foregone conclusion because once the software code was classified as printed matter, it could be disregarded in the obviousness determination. The only remaining element in the claim was a prior art storage device, which was, of course, obvious.

87. In its appeal to the Board, IBM objected to the PTO's application of the printed matter doctrine

and argued that *Diehr* required consideration of the claim as a whole, rather than focusing solely on old elements of the claim after removal of the novel elements through the application of a judicially-created exception to eligibility. IBM analogized the claimed article to a novel cam or gear. Upon insertion into a computer, the software code on the claimed diskette functions to control the machine's operation in a novel and nonobvious manner. This is similar to the conclusion in Judge Rich's majority opinion in *Alappat* that implementation of novel software code transforms a conventional computer into a novel, special purpose machine. Of course, not every element of a machine is sufficiently novel and unobvious to be patentable, and machining a new profile on a prior art cam does not represent patentable innovation without consideration of the profile's novel effect upon the machine it controls.[134]

88. The PTO's application of the printed matter doctrine to deny patent eligibility to the article of manufacture claims is best classified as a cheat. The PTO wanted to avoid the daunting task of applying novelty, nonobviousness, and enablement requirements to software claims containing no structural limitations. Rather than relying solely upon application of the printed matter doctrine, the Board should have addressed the implications of allowing such article of manufacture claims.

89. Ten amicus briefs were filed in the *Beauregard* appeal to the CAFC, nine urging reversal of the Board's rejection and one taking no formal position. The software industry strongly protested the PTO's rejection of the claims in *Beauregard* in part because, after the *Alappat*, *Warmerdam*, and *Lowry* decisions, it seems clear that the claim at issue was directed to eligible subject matter.[135]

90. The *Lowry* decision, ruling that the printed matter doctrine was inapplicable to machine-readable software code, created a dilemma for the PTO. The Board's only justification for its rejection of Beauregard's claims was the printed matter doctrine. Rather than defending a dead letter, the PTO filed a motion requesting remand of the case to the Board so that a new opinion could be crafted without reliance on the printed matter doctrine.

91. The CAFC first denied this motion,[136] but then one month later granted the motion to remand citing the PTO Commissioner's statement, "that computer programs embodied in a tangible medium, such as floppy diskettes, are patentable subject matter under 35 U.S.C. § 101 and must be examined under 35 U.S.C. §§ 102 and 103."[137]

92. In granting the motion to remand, the CAFC clearly ruled that there is no *per se* § 101 bar to such claims. Unfortunately, the applicability of the printed matter doctrine was the sole controversy in *Beauregard*. Thus, the CAFC did not have an opportunity to rule on the applicability of the 'rule of reason' patentability requirements to such claims, i.e., the proper application of §§ 102, 103, and 112. It is with respect to these requirements that *Beauregard*-type article of manufacture claims raise troubling questions.

B. PTO Examination Guidelines

93. Subsequent to the remand of *Beauregard*, the PTO issued guidelines for the examination of article

of manufacture claims to software inventions.[138] The PTO has attempted to maintain the prohibition on the patenting of information and abstract ideas, embodied earlier in the printed matter doctrine, by classifying ‘descriptive material’ as either functional or non-functional. Functional descriptive material is defined as, "data structures and computer programs which impart functionality when encoded on a computer-readable medium."[139] After *Beauregard*, the PTO has deemed such material, as stored on a computer readable medium, eligible for patent protection if it satisfies the other patentability requirements, including §§ 102, 103, and 112. In accordance with the *Warmerdam* decision, functional descriptive material *per se*, i.e., claimed without the interacting structure of the computer-readable medium, would be rejected as non-statutory because of its independence from physical limitations.[140]

94. The other category of descriptive material, non-functional, is defined as information that even when stored on a computer readable medium has no functional purpose or interrelation to any physical structure.[141] This category includes data representing creative or artistic expression.[142] Non-functional descriptive material is never eligible for patent protection because the novel aspects of such material are already protected by copyright laws.[143]

95. The Guidelines provide a useful framework for determining the patent eligibility of software inventions under § 101 that achieves reasonable accord with the CAFC case law. Unfortunately, proper application of the novelty, nonobviousness, and enablement requirements contained in the remainder of the patent statute is far more difficult to determine. There is no guidance from the CAFC on these issues, and the Guidelines merely state that the analysis for compliance with those sections is the same as for inventions in any field of technology.[144] As discussed *infra*, article of manufacture claims raise difficult questions of compliance with enablement and description requirements per § 112, and nonobviousness per § 103.

C. Advantages of Beauregard-type Claims

96. The arguments in favor of allowing patents on *Beauregard*-type article of manufacture claims are a direct result of the ease with which such claims can be asserted directly against the producer or distributor of infringing software. Most importantly, in regard to such claims the activities of the end user, i.e., the consumer, are no longer relevant to a finding of direct infringement.

97. Prior to article of manufacture claims for software as contained on a computer-readable substrate, software inventions had to be claimed in terms of the algorithm process as implemented through machine hardware. Such claims, though, are only infringed when the software is implemented by the end user through the claimed hardware, or its equivalents. The production and distribution of the software element of such claims are not sufficient to find infringement. The only way to assert such patent rights is to bring suit against the end-user because only they practice the invention as claimed, i.e., including the implementing hardware elements of the claim.[145]

98. However, patentees are usually reluctant to sue consumers of the infringing software because those consumers are often customers of the patentee as well.[146] Instead, the patent holder may

bring suit for contributory infringement against the software producer and distributor under § 271(c).[147] Evidence of direct infringement by the end user is necessary for proof of contributory infringement, and the patentee must also prove that the alleged contributory infringer knew that the combination for which its components were specifically made was both patented and infringing.[148] Therefore, damages for contributory infringement are awarded only for acts occurring after the contributory infringer had knowledge of the patent.[149]

99. Allowing claims to software contained on a computer-readable memory device as an article of manufacture eliminates the burden of proceeding under a contributory infringement cause of action. The patent holder is instead free to bring suit for direct infringement against the software producer and/or distributor without joining customers as co-defendants or satisfying additional burdens of proof.[150] This results in the same measure of patent protection on software innovations as has always been available on hardware innovations.

100. Modern computer hardware and software are completely interchangeable, in the sense that any combination of hardware and software may be used to implement any particular algorithm.[151] The decision whether to implement an algorithm using hardware or software should be based solely upon considerations such as cost, speed, and reliability. Allowing greater patent protection for the hardware implementation of an invention than its equivalent software implementation distorts the decision between hardware and software implementations, decreasing allocative efficiency.[152] *Beauregard*-type claims increase allocative efficiency by eliminating the disparity in patent protection between software and hardware implementations.

101. Finally, *Beauregard*-type article of manufacture claims have the potential to greatly expand the use of object-oriented programming. Currently, most software engineers craft original program code for each new application program. This technique is an inefficient and error-prone process, similar to the state of manufacturing prior to the advent of standardized, interchangeable parts.[153] Every software program is actually a collection of algorithm components working in concert. These algorithm components are tailor-made for each application. But, if patent protection were available for such software components, as it is under *Beauregard*, they could be marketed in the same way as hardware components.

102. Hand-crafted, application-specific software algorithms are notoriously unreliable.[154] The availability of patent protection for software algorithm components would encourage software companies to perfect and market standardized algorithm components, referred to as software 'objects' in the art. These software objects would then be combined and organized by software engineers to create novel application programs in the same way that mechanical design engineers combine standardized fasteners and materials to create novel machines. This advance could greatly reduce software development costs and improve software reliability and performance.[155]

D. Disadvantages of Beauregard-type claims

103. As shown above, *Beauregard*-type claims have one substantial advantage over previous types of software patent claims, streamlined enforcement. On the other hand, in many ways such claims are inconsistent with the case law on the patenting of algorithm inventions. In addition, the examination guidelines used by the PTO to evaluate patent applications containing *Beauregard*-type claims are based upon distinctions that have no foundation in software art. Further, the PTO's allowance of means-plus-function language with inadequately supporting disclosure in the patent specification has created serious § 112 enablement and description concerns.

1. Problem #1: The (Lack of) Distinction Between Functional and Nonfunctional Information.

104. One interpretation of the doctrinal justification for allowing *Beauregard*-type article of manufacture claims, while denying patentability to articles such as musical CDs, is that the information protected by a *Beauregard*-type claim is functional in that it has the ability to directly control the computer used to implement the patented algorithm.[156] In the wake of *Beauregard*, the PTO's examination guidelines have attempted to maintain this distinction between unpatentable information, such as a musical performance encoded on a CD, and patentable software instruction by classifying descriptive material encoded in a computer readable medium as either functional, i.e., patentable, or non-functional, i.e., nonpatentable. But, this distinction finds no support in the actual architecture of the implementing computer hardware:

> Executable memory differs from a program repository because it implicitly includes the digital computer processor that it directly controls. Functionality is imparted to the controlled digital computer only from information stored in executable memory.
>
> In contrast, programs in a repository memory, such as a hard disk or floppy disk, are merely data for another computer program which loads them into the executable memory. The programs do not configure the computer until they have been loaded into executable memory. In most cases, a loader program must transform the programs into the form that can be executed by the computer (e.g., linking to libraries, relocating addresses, and allocating data areas). There is little difference between the digital data representing programs stored in a repository memory and the digital data representing the audio stored on a compact disc. Both are data that are read and processed digitally to bring about the desired results. Information stored in repository memory does not directly impart any functionality to a computer system.
>
> It is important that law be anchored in reality, and distinctions be drawn only where they actually exist. The "mathematical algorithm" distinction did not work well because there is no similar distinction in computer science. In contrast, the concept that a general purpose computer when programmed becomes a new machine works well because the idea matches almost exactly the concept of virtual machines discussed in operating systems classes.

> The Examination Guidelines make a distinction not anchored in computer science. Rather than recognize that the only memory where information is "functional" is executable memory, such as RAM or ROM, the Examination Guidelines try to include repository memory as being able to hold functional information. However, the information stored in a repository memory does not impart functionality to a digital computer until it has been loaded into executable memory. Until then, it is simply data, no different than "non-functional descriptive material" such as music, literary works, or a compilation of factual data stored in the same medium.[157]

105. The problem described above is troubling because basing patent eligibility on unrealistic distinctions and inaccurate classifications will impair the ability of the PTO and the courts to provide patent protection for software inventions that is both useful and consistent with the patent doctrine in other technologies. The PTO's functional/non-functional classification schemes will result in both Type I and Type II errors precisely because it is not based upon realistic distinctions.

a. The Solution: Amend the Patent Statute

106. The clunky fiction of claiming software as a "program stored in memory" would be unnecessary if the patent statute were amended to classify as direct infringement the distribution of a computer program which, upon execution, implements a patented method.[158] This relatively simple amendment would eliminate the need for superfluous prior-art structural limitations in software claims. Such claims could once again be crafted in terms of process and method limitations, which more accurately represent software innovation.

107. A Tennessee District Court applied this doctrine in a non-software case, holding that selling a patented typewriter ribbon cassette was direct infringement, even though the cassette had been claimed in relation to other parts of the typewriter.[159] The cassette was the sole novel feature of the claim. The court ruled that because the other elements of the claim were present solely to help define the cooperation between the elements during use, they were not part of the claimed invention. The CAFC affirmed without opinion.

108. *Smith Corona* has been interpreted by some to stand for the proposition that the prior-art structural limitations included in software claims in order to insure § 101 eligibility may be disregarded in determining direct infringement.[160] This is just not so.

109. The claims in *Smith Corona* were poorly drafted. The only reason that the prior art elements were included in the claim with the novel cassette mechanism was to indicate the parts that the invention was designed to mate with during use. The court held, therefore, those prior art elements could be disregarded without affecting the eligibility of the claim.[161]

110. Section 117 of the Copyright Act permits copying of a computer program if such copying is a necessary step in its use.[162] The patent laws contain no equivalent rule and, therefore, the user of any software program protected by a *Beauregard*-type claim would necessarily infringe the

patent, both upon installing the program from floppy disk or CD-ROM onto the computer's hard disk and again each time the program is loaded from the hard disk into executable memory in preparation for use.[163]

111. Admittedly, this is not a major problem. Still, "program stored in memory" claims necessitate amendment of the patent laws to include provisions for the user rights contained in the Copyright Act, particularly those in § 117. Without such amendment, *Beauregard*-type claims represent a distortion of both the patent and copyright systems simply to provide an easy method to assert patent rights against the producers, rather than the end users, of infringing software.[164]

2. Problem #2: Indefinite Claiming in Violation of § 112, ¶ 2.

112. The claims allowed in *Beauregard* are either indefinite or unenforceable under the second paragraph of § 112.[165] The *Beauregard* claims inadequately describe the environment in which the patented program will implement the claimed function, in violation of the requirement in paragraph two of § 112 that the claims particularly point out and distinctly claim the subject matter regarded as the invention.[166]

113. Suppose that a computer program patent includes a "program stored in memory" claim to a process comprising three means-plus-function elements A, B, and C, with no description of the implementing hardware. It is possible that in implementing the program, some hardware platforms would substitute a different, perhaps more efficient, algorithm element such as A* in the place of A. This implementation would not be covered by the patent because the infringing condition, i.e., implementation of process steps A, B, and C was never obtained.[167] Patent infringement would be hardware dependent and, therefore, beyond the control of the allegedly infringing software distributor.

114. If there are hardware platforms on which the allegedly infringing software may be implemented without infringement, it is illogical to find direct infringement for the mere possession of the allegedly infringing software.[168] Once again, infringement is dependent upon the actions of the end user--the condition that *Beauregard*-type claims were designed to avoid. This uncertainty of infringement suggests that such claims are indefinite under § 112 because they fail to reasonably apprise those of skill in the art of their scope.[169]

a. The Solution: Detailed Disclosure of Implementing Hardware

115. By requiring that *Beauregard*-type program product claims include a reasonably detailed disclosure of the ancillary software and hardware platforms intended for implementation, the scope of protection for such claims will be more readily discernible.[170] Viewed in isolation, this will prevent the extension of the scope of protection beyond such claims, but only to the extent necessary to satisfy § 112's requirement that the patent disclosure particularly point out and distinctly claim the patented invention.

3. Problem #3: Consistency with Precedent

116. The only practical use of an algorithm such as that claimed in *Beauregard* is through software encoding for implementation by computer. If a patent is allowed on the invention claimed as software code, then there is no practical method to practice the algorithm without infringing the patent. Therefore, article of manufacture claims such as those in *Beauregard* are, in practical effect, claims to the algorithm itself and therefore violative of the basic policy against patents on algorithms *per se*.[171] Unfortunately, neither the Board's rejection nor the CAFC's remand order addressed this problem.

a. Solution: Reinstate the Requirement for Meaningful Structural Claim Limitations

117. Solution of this problem is mutually exclusive with the validity of "program in memory" article of manufacture claims. Insuring that all practical use of an algorithm is not foreclosed requires the inclusion of meaningful structural limitations as elements of the claim. Although this would make software patents harder to enforce, as discussed above, it would rescue the long-standing restriction on claims to abstract ideas that seems to have been sacrificed to pragmatism with *Beauregard*.

IX. Sui Generis Protection of Algorithmic Innovation

118. Rather than attempting to amend the current patent statute, many commentators have suggested enacting a separate protection scheme for algorithm inventions. Many features of the current patent law, e.g., in-depth and time-consuming examination prior to issue, are detrimental to the effective protection of software invention. A scheme for algorithm protection separate from the patent statute could be enacted to provide protection specifically tailored to the needs of software innovators.

119. Richard Stern has proposed such a 'petty' patent statute specifically designed for the protection of algorithms.[172] Under his proposal, algorithm inventions would be eligible for protection under a newly added Part V of Title 35. The PTO would issue petty patents after a cursory examination to ensure compliance with statutory compliance and regulations. The PTO would not examine such applications for validity; this would be left to the courts in the unlikely event of litigation. Therefore, acquiring software patent protection would be faster and less expensive than under the current patent statute.

120. Some other common features of such *sui generis* proposals are a reduction of the novelty and nonobviousness requirements, and a likewise reduction in the term of protection.[173] One disadvantage of such a retreat from the current patent statute would be the necessity of enacting duplicative international treaties recognizing and supporting such a protection scheme.[174]

X. Conclusion

121. The patent eligibility of article of manufacture claims to software invention as embodied in a computer-readable memory device is uncertain. The distinctions used by the PTO to determine whether descriptive material, i.e., data on computer disk or CD-ROM, is eligible for patent protection have no foundation in the actual properties of such data. When the distinctions used to determine patent eligibility are not founded in reality, such determinations become unpredictable and unprincipled, as was the case under the now defunct algorithm exclusion.

122. The only guidance provided by the courts with respect to such claims is that the printed matter doctrine is inapplicable. The PTO is currently allowing such article of manufacture claims having no limitations, and often no description, of the implementing hardware. The scope of these claims is uncertain, in violation of § 112. In acquiescing to allow such claims, the PTO has needlessly delayed the consideration of many difficult validity issues, such as nonobviousness. These issues have been left to the trial courts where infringement suits based on such claims will be decided in a predictably chaotic manner.

123. The fundamental problem with such claims, and in fact the reason for their existence, is that the current patent statute is woefully inadequate for the protection of computer software. *Beauregard*-type article of manufacture claims represent an ad hoc, partial solution to this problem. A better solution, and the only one that will result in useful and predictable patent protection of software, would be guidance from the legislature. The courts, the PTO, and practitioners need better tools for software protection in the form of amendments to the current patent statute, or a new statute specifically designed to protect the intangible innovation as embodied in computer software. The policy decisions inherent in protecting software are simply too complicated to allow adequate solution through the CAFC's legislating from the bench.

Footnotes

[*] Registered Patent Agent #42,861; J.D. candidate 1999, The University of Virginia School of Law; M.B.A., The College of William & Mary; B.S., Mechanical Engineering, Old Dominion University.

[1] *See, e.g.*, Richard H. Stern, *Solving the Algorithm Conundrum: After 1994 in the Federal Circuit Patent Laws Needs a Radical Algorithmectomy*, 22 AIPLA Q.J. 167, 169 (1994).

[2] *See* David Abraham, *Suggestions for Improved Intellectual Property Protection of Software, or Where is Alexander When You Really Need Him?*, 23 S.U.L. Rev. 293 (1996).

[3] *See* Stern, *supra* note 1, at 170.

[4] *See id*.

[5] *See id*.

[6] *See* Lee A. Hollaar, *Justice Douglas was Right: The Need for Congressional Action on Software Patents*, 24 AIPLA Q.J. 283, 286 (1996).

[7] *See id*.

[8] *See id*. Unfortunately, the PTO's minimal disclosure requirements in combination with the availability of means-plus-function language for software algorithm claims has resulted in the issuance of patents that add little if nothing to the public domain, but which may be interpreted as affording a broad right of exclusion to the patent holder. *See* discussion *infra* ¶ 115.

[9] *See* Robert C. Laurenson, *Computer Software "Article of Manufacture" Patents*, 77 J. PAT. [& TRADEMARK] OFF. SOC'Y 811, 818 (1995) ("Under the abstraction/ filtration/comparison test, which has been adopted in various forms by both the Ninth and Second Circuits as the appropriate method of defining the protectable aspects of computer software under the copyright laws, the ideas behind (or functions performed by) software are not protectable under the copyright laws. . . . Thus, the scope of protection available is narrow and undefined.").

[10] *See* Hollaar, *supra* note 6, at 289 ("The result, in decisions like *Whelan* [*Assoc.*] *v. Jaslow* [*Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986)] and the final district court decision in *Lotus* [*Dev. Corp.*] *v. Borland* [*Int'l, Inc.*, 831 F. Supp. 223 (D. Mass. 1993), rev'd, 49 F.3d 807 (1st Cir. 1995), aff'd, 116 S. Ct. 804 (1996)], was an attempt to extend copyright protection to cover the 'non-literal' aspects of a program. This can result in protection beyond that provided by the patent law, without disclosure or examination, and for a much longer period of time. But a working patent system for software-related inventions can remove much of the need for protecting software by copyright beyond literal or near-literal copying, resulting in less stretching of copyright protection.").

[11] *See* IRVING KAYTON, 1 PATENT PRACTICE 5.19 (PRG 6th ed. 1995).

[12] *See id.* In the first case after this period in which a patent was upheld as valid by the Eighth Circuit, the Court of Appeals had to overrule the trial judge's 'finding' of invalidity. In his opinion the trial judge wrote that even though nonobviousness under § 103 had been established, he felt obligated to strike the patent as invalid based on the Eighth Circuit's history of holding all patents invalid. *See Woodstream Corp. v. Herter's, Inc.*, 312 F. Supp. 369 (D. Minn. 1970).

[13] For an enlightening discussion of the process by which elected legislators shift the burden of, and accountability for, difficult policy decisions to appointed administrative agencies through the enactment of vaguely worded statutes see JOHN HART ELY, DEMOCRACY AND DISTRUST: A THEORY OF JUDICIAL REVIEW

131-34 (1995).

[14] But one of the hallmarks of innovation is that by its very nature it cannot be anticipated. S*ee, e.g.*, *Diamond v. Chakrabarty*, 447 U.S. 303, 316 (1980) ("A rule that unanticipated inventions are without protection would conflict with the core concept of the patent law that anticipation undermines patentability.").

[15] For example, the printed matter doctrine was originally developed as a *per se* rule against the patenting of articles whose sole novelty consisted of imprinted expression (protection for which was thought preferable through copyright law).

[16] *Arrhythmia Research Tech., Inc. v. Corazonix Corp.*, 958 F.2d 1053, 1061-66 (Fed. Cir. 1992).

[17] S. Rep. No. 82-1979, at 5 (1952), *reprinted in* 1952 U.S.C.C.A.N. 2394, 2399.

[18] 56 U.S. (15 How.) 62 (1853).

[19] *Id.* at 112, *quoted in* Stern, *supra* note 1, at 171.

[20] *See id.* at 72; *see also* Stern, *supra* note 1, at 171.

[21] *See O'Reilly*, 56 U.S. (15 How.) at 113.

[22] *See* Stern, *supra* note 1, at 171. Of course, Morse's claim 8 would have expired long before the invention of the fax machine, but during its period of efficacy it would have prevented the independent efforts of others to either improve upon Morse's method, or to independently develop other methods of electromagnetic communication. *See id.* at 172.

[23] *See id.*

[24] *See* 35 U.S.C. § 112 (1988).

[25] *See* Stern, *supra* note 1, at 172. The bar to patent eligibility for laws of nature such as gravity and electromagnetism is imputed to 35 U.S.C. § 101 (1988), and after *Diamond v. Diehr* and *Diamond v. Chakrabarty*, both discussed *infra*, this remains as one of the few § 101 *per se* bars to patent eligibility.

[26] *See id.*

[27] In *Crystals and Mud in Property Law*, Carol Rose discusses the pros and cons of applying *per se* rules rather than performing an individualized analysis based upon the specific facts in a given situation.

Carol M. Rose, *Crystals and Mud in Property Law*, 40 Stan. L. Rev. 577 (1988). Rose argues that the common law of property has oscillated between 'crystal' *per se* rules, which are useful in providing clear signals regarding the allocation and protection of valued property, and 'mud' rules based upon a balancing of the equities in each individual situation, which are useful in avoiding hardship to those who are unable to function within the framework of the 'crystal' *per se* rules. *See id*. at 590-93. Rose proposes that stable equilibrium obtains under neither the crystal nor the mud rules because of society's mutually exclusive desires for low transaction costs and the avoidance of inequitable individual hardship. *See id.* at 593-95.

[28] 409 U.S. 63 (1972).

[29] *See id*. at 73.

[30] *Id*. at 71, 73.

[31] *See* Martin J. Adelman et al., Patent Law 111 (1998).

[32] *See Benson*, 490 U.S. at 71-72.

[33] *See* Adelman et al., *supra* note 31, at 112.

[34] *Cf. id.* (questioning whether the Court's finding of ineligibility under § 101 may have been at least partially due to the fact that they were simply not very impressed with the claimed invention).

[35] *Benson*, 490 U.S. at 67; *but see* Adelman et al., *supra* note 31, at 132 ("How legitimate is this rationale? Should patents be refused to inventors of otherwise patentable microscopes, voltmeters, centrifuges and other sorts of everyday equipment in the laboratories of contemporary scientists, engineers and technicians?").

[36] Stern, *supra* note 1 at 174 (quoting *Gottschalk v. Benson*, 409 U.S. 63, 70 (1972)).

[37] *See id.*

[38] 437 U.S. 584 (1978).

[39] *See id*. at 586.

[40] *See id*. at 591-92 ("The process itself, not merely the mathematical algorithm, must be new and useful. Indeed, the novelty of the mathematical algorithm is not a determining factor at all. . . . [I]t is treated as though it were a familiar part of the prior art.").

[41] *Id.* at 594.

[42] *See* ADELMAN ET AL., *supra* note 31, at 114.

[43] *See* Keith E. Witek, *Developing a Comprehensive Software Claim Drafting Strategy for U.S. Software Patents*, 11 BERKELEY TECH. L.J. 363, 374 (1996).

[44] *See id.* at 375 ("Most software patent practitioners of this period continued to draft apparatus and structure claims for software inventions which appeared at first glance to be hardware inventions in order to avoid an adverse 35 U.S.C. § 101 ruling. This response from patent attorneys was exactly what the Court was trying to avoid.").

[45] 450 U.S. 175 (1981).

[46] *See* Witek, *supra* note 43, at 377.

[47] *See Diehr*, 450 U.S. at 184.

[48]

> In determining the eligibility of respondents' claimed process for patent protection under §101, their claims must be considered as a whole. It is inappropriate to dissect the claims into old and new elements and then to ignore the presence of the old elements in the analysis. This is particularly true in a process claim because a new combination of steps in a process may be patentable even though all the constituents of the combination were well known and in common use before the combination was made.

See id. at 188. This is directly adverse to the holding in *Flook*.

[49] *See id*. at 191.

[50] *See* ADELMAN ET AL., *supra* note 31, at 144.

[51] *See id.*

[52] 22 F.3d 290 (Fed. Cir. 1994).

[53] 33 F.3d 1526 (Fed. Cir. 1994).

[54] 33 F.3d 1354 (Fed. Cir. 1994).

[55] 32 F.3d 1579 (Fed. Cir. 1994).

[56] 42 F.3d 1376 (Fed. Cir. 1994).

[57] *See, e.g.*, Stern, *supra* note 1.

[58] *See Schrader*, 22 F.3d at 291 (The claimed algorithm was a method for determining the combination of auction bids to be selected as winners for a variably divisible article for sale, in this case a parcel of land that could be subdivided. The algorithm analyzed all bids submitted and determined which group of bids to select as 'winners' so that: (1) none of the selected bids were for overlapping subdivisions, and (2) sales proceeds for the aggregate parcel of land were maximized.); *see also* Stern, *supra* note 1, at 179.

[59] *See Schrader*, 22 F.3d at 293-94.

[60] *See id.* at 294.

[61] *See id.* at 292. The Freeman-Walter-Abele (FWA) test is a two-step determination of patent eligibility for claims containing algorithm elements. The FWA test was developed by the Court of Appeals for the Federal Circuit's predecessor, the Court of Customs and Patent Appeals (CCPA), and was arguably invalidated by the *in banc* ruling in *Alappat*, as indicated by the subsequent withdrawal of the decision in *Trovato* where the FWA test had been applied resulting in a determination of ineligibility. *See* ADELMAN ET AL., *supra* note 31, at 171.

[62] *Schrader*, 22 F.3d at 293.

[63] *See id.* at 292.

[64] *See id.* at 293.

[65] *See id.* at 293-94. The claims did include the step of entering the bids in a 'record,' but this physical element was dismissed as insufficient to impart patentability based upon the decision in *In re Grams*, 888 F.2d 835 (Fed. Cir. 1989), because it was indistinguishable from data gathering steps. *See Schrader*, 22 F.3d at 294.

[66] *See Schrader*, 22 F.3d at 294.

[67] *See Arrhythmia*, 958 F.2d 1053; *see also In re Abele*, 684 F.2d 902 (CCPA 1982) (holding that claims involving the manipulation of data representing CAT scan images were eligible under § 101).

[68] *Schrader*, 22 F.3d at 294.

[69] *Id.*

[70] *Id.* at 293-94.

[71] Stern, *supra* note 1, at 182.

[72] *Trovato*, 42 F.3d at 1383.

[73] *See Schrader*, 22 F.3d at 296.

[74] *See id*. at 297.

[75] *See Diehr*, 450 U.S. 175; *see also Diamond v. Chakrabarty*, 447 U.S. 303 (1980).

[76] *See Arrhythmia,* 958 F.2d 1053; *see also Abele*, 684 F.2d 902.

[77] *Schrader*, 22 F.3d at 298.

[78] *See id*.

[79] *See* Abraham, *supra* note 2, at 293 (citing *In re Alappat*, 33 F.3d at 1583 (Rader, J., concurring)).

[80] *See id*.

[81] *Hilton Davis Chem. Co. v. Warner-Jenkinson Co., Inc.*, 62 F.3d 1512 (Fed. Cir. 1995), *rev'd on other grounds*, 520 U.S. 17 (1997).

[82] *See* Stern, *supra* note 1, at 182 (citing *Malta v. Schulmerich Carillons, Inc.*, 959 F.2d 923 (Fed. Cir. 1992)).

[83] *See* Stern, *supra* note 1, at 183.

[84] *Alappat*, 33 F.3d at 1540-41.

[85] Claim 15 reads:

> A rasterizer for converting vector list data 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 the vertical distance between the endpoints of each of the vectors in the data list;
> > (b) means for determining the 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.

Id. at 1538-39.

[86] *See id.* at 1544.

[87] *See id.* at 1537.

[88] *See id.* at 1583 (Rader, J., concurring).

[89] *See id.* at 1544.

[90] *See* Stern, *supra* note 1, at 185.

[91] *See id.* at 186.

[92] *Alappat*, 33 F.3d at 1545 (citations omitted).

[93] *See* Stern, *supra* note 1, at 188.

[94] *See Alappat*, 33 F.3d at 1564-65 (Archer, Chief J., dissenting).

[95] *See id.* at 1564.

[96] *Id.* at 1563-64 (emphasis added; italics and footnote omitted).

[97] *See* 35 U.S.C. §§ 102, 103 (1988).

[98] Judge Learned Hand recognized this assertion as early as 1935 when he stated:

> [D]efendant argues that the supposed invention is no more than a substitution of materials familiar to the art in the same uses; an aggregation of which each part performs what it did before. We may concede as much arguendo, for the same may be said of every invention. All machines are made

> up of the same elements; rods, pawls, pitmans, journals, toggles, gears, cams, and the like, all acting their parts as they always do and always must. All compositions are made of the same substances, retaining their fixed chemical properties. But the elements are capable of an infinity of permutations, and the selection of that group which proves serviceable to a given need may require a high degree of originality. It is that act of selection which is the invention.

B.G. Corp. v. Walter Kidde & Co., Inc., 79 F.2d 20, 21-22 (2d Cir. 1935).

[99] *Alappat*, 33 F.3d at 1565.

[100] There is a subtle, although fundamental, difference between determination of equivalent structure for interpreting the scope of means-plus-function claims under § 112 ¶ 6, and determining whether infringement has occurred under the Doctrine of Equivalence. Determining equivalent structure of means-plus-function claims is a legal process controlled by statute (§ 112 ¶ 6) by which the scope of a claim is determined. In adjudging infringement under the equitable Doctrine of Equivalence, the scope of the claim is invariant, rather the court must determine whether an alleged infringing article or process falls within the patent holder’s exclusive right, which extends beyond the strict construction of the claim to include protection against articles which are "insubstantially different" from the invention as claimed. Despite these differences, the CAFC determines proper application of both doctrines, and its rulings regarding one doctrine affect the other as well. *See generally* KAYTON, *supra* note 12, at 2.25.

[101] *Warmerdam*, 33 F.3d 1354.

[102] *Lowry*, 32 F.3d 1579.

[103] *Trovato*, 42 F.3d 1376

[104] *See* Stern, *supra* note 1, at 192-93.

[105] *Warmerdam*, 33 F.3d at 1360-62.

[106] *See id.*

[107] *Schrader*, 22 F.3d at 293-94.

[108] *Warmerdam*, 33 F.3d at 1360.

[109] *See Lowry*, 32 F.3d at 1582-83.

[110] *In re Beauregard*, 53 F.3d 1583, 1584 (Fed. Cir. 1995).

[111] *Lowry*, 32 F.3d at 1583-84.

[112] *Alappat*, 33 F.3d at 1545.

[113] 42 F.3d 1376 (Fed. Cir. 1994).

[114] *See id.* at 1377-78.

[115] *Id.* at 1380.

[116] *Id.* at 1383 (internal quotation marks and citation omitted).

[117] *Id.*

[118] *Alappat*, 33 F.3d at 1545 (citations omitted).

[119] *See* ADELMAN ET AL., *supra* note 31, at 149.

[120] *See In re Trovato*, 60 F.3d 807 (Fed. Cir. 1995).

[121] *See generally* KAYTON, *supra* note 11.

[122] *See* Laurenson, *supra* note 9, at 814-15.

[123] *See id.* at 813.

[124] This is a bit misleading. While a *per se* rule will invariably increase the combined occurrence of Type I and Type II errors, a *per se* rule may be crafted which completely eliminates errors of one type. For example, the *per se* rule against patenting algorithms attributed to § 101 completely eliminates Type I error in situations where an applicant claims what is in fact an algorithm, i.e., there are no false positives. On the other hand, such a *per se* rule may greatly increase Type II error through misapplication, i.e., there may be a great many false negatives in which a patentable invention is denied protection through the improper application of the *per se* rule.

[125] *See generally* Rose, *supra* note 27.

[126] *See generally* Ely, *supra* note 13.

[127] *See* discussion *supra* ¶¶ 71-72.

[128] *See, e.g.*, *Warmerdam*, 33 F.3d at 1360.

[129] *Lowry*, 32 F.3d at 1584.

[130] In the case of software applications used by large businesses, it is possible that the infringing end-users would also be customers of the patent holder, creating additional barriers to enforceability.

[131] 53 F.3d 1583 (Fed. Cir. 1995).

[132] *See* Stern, *supra* note 1, at 196 n.103.

[133] *See In re Miller*, 418 F.2d 1392, 1396 (CCPA 1969); *In re Jones*, 373 F.2d 1007, 1012 (CCPA 1967).

[134] *See* Stern, *supra* note 1, at 198.

[135] *See* Laurenson, *supra* note 9, at 819-20.

[136] *See In re Beauregard*, Fed. Cir. App. No. 95-1054 (Order of Apr. 6, 1995).

[137] *Id*. at 1584.

[138] *See* Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478 (1996).

[139] *Id*. at 7481, *quoted in* Ruben Bains, *A Comparison of the PTO's Computer-Implemented Guidelines with the Current Case Law*, 5 Tex. Intell. Prop. L. J. 27, 33 (1996).

[140] *See* Bains, *supra* note 139, at 34.

[141] *See id*.

[142] *See id*.

[143] *See id*. at 35.

[144] *See* Examination Guidelines, *supra* note 138, at 7487, *construed in* Bains, *supra* note 139, at 45.

[145] *See* Brief for Amicus Curiae The Philadelphia Intellectual Property Law Association at 14;

Beauregard (No. 95-1054).

[146] *See id*. at 13.

[147] 35 U.S.C. § 271(c) (1988).

[148] *See* Brief for Amicus, *supra* note 145, at 15.

[149] *See id*.

[150] *See id*.

[151] *See* Stern, *supra* note 1, at 209-10.

[152] *See id*. at 210.

[153] *See* W. Wayt Gibbs, *Software's Chronic Crisis*, SCI. AM., Sept. 1994, at 86, 87 ("[S]oftware engineering remains a term of aspiration. The vast majority of computer code is still handcrafted from raw programming languages by artisans using techniques they neither measure nor are able to repeat consistently. 'It's like musket making was before Eli Whitney,' says Brad J. Cox, a professor at George Mason University.").

[154] In *Software's Chronic Crisis*, Gibbs describes what has probably been the highest-profile software debacle in history. Gibbs, *supra* note 153, at 86. The Rube Goldberg-esque automated baggage routing system at Denver International Airport has been plagued by software control errors so numerous and severe that the airport's opening was delayed by more than nine months. *See id.* In a striking display of the unpredictable reliability of customized software, Pittsburgh International Airport implemented a very similar automated baggage handling system with no major software problems.

[155] *See id*. at 91 (examining Raytheon's savings of $17.2 million in software development costs over a six-year period through the use of object-oriented programming techniques).

[156] *See Lowry*, 32 F.3d at 1583 (ruling that the printed matter doctrine has no relevance to the claimed computer memory structure because it is processed by a machine).

[157] Hollaar, *supra* note 6, at 293-94.

[158] *See id*. at 297.

[159] *See Smith Corona Corp. v. Pelikan, Inc.*, 784 F. Supp. 452, 460-63 (M.D. Tenn. 1992), *aff'd*, 1 F.3d

1252 (Fed. Cir.), *reh'g denied, suggestion for reh'g in banc declined* (Jul. 19, 1993).

[160] *See* Edward P. Heller, III, *Letter to the Editor*, 78 J. Pat. & Trademark Off. Soc’y 188, 189 (1996).

[161] *See Smith Corona*, 784 F. Supp. at 465.

[162] *See* 17 U.S.C. § 117 (1994).

[163] *See* Hollaar, *supra* note 6, at 295-96.

[164] *See id*. at 297.

[165] *See* Steven W. Lundberg & Russell D. Slifer, *"Program Product" Patents: Savior or Blunder?*, 12 No. 8 Computer Law. 1, 2 (1995).

[166] *See id*. at 3; *see also* 35 U.S.C. § 112 (1994).

[167] *See* Lundberg & Slifer, *supra* note 165, at 4.

[168] *See id*.

[169] *See id*. (citing *Warmerdam*, 33 F.3d 1354).

[170] *See generally id*. at 5.

[171] *See* Stern, *supra* note 1, at 199.

[172] *See id*. at 213.

[173] *See* Adelman et al., *supra* note 31, at 152.

[174] *See id*. (quoting The Advisory Commission on Patent Law Reform: A Report to the Secretary of Commerce 151-52 (1992)).