

The John Marshall Journal of Information Technology & Privacy Law

Volume 17
Issue 1 *Journal of Computer & Information Law*
- Fall 1998

Article 5

Fall 1998

Patentability of Computer Software Instruction as an "Article of Manufacture": Software as Such as the Right Stuff, 17 J. Marshall J. Computer & Info. L. 89 (1998)

Vincent Chiappetta

Follow this and additional works at: <https://repository.law.uic.edu/jitpl>



Part of the [Computer Law Commons](#), [Intellectual Property Law Commons](#), [Internet Law Commons](#), [Privacy Law Commons](#), and the [Science and Technology Law Commons](#)

Recommended Citation

Vincent Chiappetta, Patentability of Computer Software Instruction as an "Article of Manufacture": Software as Such as the Right Stuff, 17 J. Marshall J. Computer & Info. L. 89 (1998)

<https://repository.law.uic.edu/jitpl/vol17/iss1/5>

This Symposium is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in The John Marshall Journal of Information Technology & Privacy Law by an authorized administrator of UIC Law Open Access Repository. For more information, please contact repository@jmls.edu.

PATENTABILITY OF COMPUTER SOFTWARE INSTRUCTION AS AN “ARTICLE OF MANUFACTURE:” SOFTWARE AS SUCH AS THE RIGHT STUFF

by VINCENT CHIAPPETTA†

TABLE OF CONTENTS

I. Introduction	90
II. Background	92
III. Analysis	95
A. Origins of United States Patent Law and Relevant Patent Law Basics From Whence They Came and Where They Seek To Go	97
1. <i>The Basic Patentability Inquiry</i>	99
B. A Short History of Software Patentability Up To Article of Manufacture Claims: How the World Started to Change With <i>Alappat</i>	106
C. “Article of Manufacture” Claims in the CAFC and the PTO	114
D. The PTO Guidelines	121
E. What’s Wrong With Current Article of Manufacture Practice? What We Are Trying to Achieve With the Patent System	127
1. <i>The Useful Arts</i>	129
2. <i>Congressional Action</i>	135
3. <i>Can the Current “Software as Article of Manufacture” Practice Get Us There?</i>	139
a. Bounding the Inquiry	139
b. Scope of the Claim Format	139
c. Understanding Software: Technology, Language or Both?	141

† Associate Professor, Willamette University College of Law. The author gratefully acknowledges a writing grant from the Oracle Corporation.

F. Software Under the Guidelines	143
1. <i>Syllogism One</i>	146
2. <i>Syllogism Two</i>	146
3. <i>Syllogism One (Over-inclusion)</i>	146
4. <i>Syllogism Two (Under-inclusion)</i>	150
G. A New Test and a New Syllogism	154
1. <i>Does it Work? Testing the New Syllogism for Consistency With Patent Law Objectives and Practicality</i>	159
a. Patentable Subject Matter	160
b. Novelty and Nonobviousness	168
c. Administrative Ramifications	176
d. Other Considerations	177
e. Overall Effect on Software Patentability Outcomes	177
IV. Conclusion	178
Appendix	179

I. INTRODUCTION

The last five years have witnessed a dramatic shift in the approach taken by the Court of Appeals for the Federal Circuit ("CAFC") and, under the CAFC's stern if somewhat incomplete guidance, the United States Patent and Trademark Office ("PTO") to the seemingly intractable problem of determining whether software inventions¹ qualify as patentable subject matter under the United States patent laws.² Beginning

1. Until very recently the discussion primarily focused on inventions which incorporate software. This article treats software "as such," whether standing alone or claimed as part of another invention. References to "software" are inclusive of all manifestations regardless of whether the software invention is expressed in a particular programming language, its functionality described in a written narrative or if it is tied to any particular hardware environment. Claims to software *per se* represent the most difficult analytical case. Any analysis which deals with that case can be equally applied to situations where software serves as a component in a more complex computer system claim.

There are related questions which arise when the claimed invention involves computer readable data. Although these data questions are outside the primary scope of this article the issue is explained and discussed in general terms below. See *infra* note 242 and accompanying text.

2. 35 U.S.C. §§ 1-376 (1994). There are a number of requirements relating to the overall patentability of inventions including novelty under 35 U.S.C. § 102 (1994), non-obviousness under 35 U.S.C. § 103 (1994) and the adequate description and enablement of the invention under 35 U.S.C. § 112 (1994). See discussion *infra* notes 41-70 and accompanying text. This article focuses primarily on the patentable subject matter requirement, is software the "right stuff" under 35 U.S.C. § 101 (1994), but also considers the interaction between the basis for making those decisions and other patentability requirements. See discussion *infra* notes 47-50, 302-311, 320-338 and accompanying text.

with a series of CAFC decisions in 1994³ and culminating with the PTO's issuance of its Final Examination Guidelines for Computer-Related Inventions (the "Guidelines") in 1996,⁴ the paradigm shifted from a "mathematical algorithm" based analytic structure⁵ to an apparatus model driven by the presence or absence of related computer hardware in the claims.⁶

Has this new approach finally brought the 30-year quest for a solution to a close?⁷ Certainly, abandoning the mathematical algorithm approach is a proper, and indeed vital, first step⁸ toward untangling the confusion surrounding the patentability of computer software. And from the applicant's and examiner's perspectives the new regime establishes a more well defined and objective set of requirements which when followed results in much greater certainty in the examination process.⁹ Unfortunately however, this promising alternative leaves the Gordian knot uncut. This article examines the current treatment of the increasingly important "article of manufacture" variant¹⁰ of the new hardware model and explains logical gaps that still remain. The results of this inquiry are used as the basis for a proposed new software as implementation versus software as language approach to resolving the software patentability conundrum.

3. See *infra* notes 95-160 and accompanying text.

4. Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7,478 (1996) [hereinafter *Guidelines*]. See *infra* notes 161-191 and accompanying text.

5. See *infra* notes 76-91 and accompanying text.

6. See *infra* notes 94-160 and accompanying text.

7. See *infra* notes 73-191 and accompanying text. The United States is not alone in struggling with this issue. See *infra* note 278 and accompanying text (discussing developments in European Union law on the subject).

8. A number of commentators have argued the mathematical algorithm approach simply does not work and should be eliminated. See, e.g., Donald S. Chisum, *The Patentability of Algorithms*, 47 U. Pitt. L. Rev. 959 (1986). I agree with Professor Chisum. See *infra* note 85 and accompanying text.

9. See *infra* note 191 and accompanying text.

10. Under the software as article of manufacture format the applicant drafts the claims to include a computer readable medium (such as a diskette or ROM) as the carrier for the computer program instructions. A set of sample claims in this format are attached herewith as an Appendix to this article. By couching the application in these apparatus terms, the applicant hopes to have an invention which primarily involves the creation of a new computer program treated as an article of manufacture. Because an article of manufacture is an expressly statutory class of patentable subject matter, this treatment virtually automatically clears the pesky § 101 examination hurdle. See *infra* notes 89-94 and accompanying text. Another common hardware claim format is to include the digital computer itself, a § 101 "machine" claim. See *infra* notes 95-123 and accompanying text. Much of the analysis of article of manufacture claims in this article also applies to these machine claims. The article of manufacture format has broader commercial appeal because of the possibility of direct enforcement against competing software distributors. See *infra* notes 92-94 and accompanying text.

II. BACKGROUND

The starting point for any software as patentable subject matter analysis must be first principles: the Constitutional policy objective of promoting "progress . . . of . . . [the] useful arts" and the nature of software itself.¹¹ In this framework¹² the normative answer to the question of "is software patentable subject matter" is straightforward enough; only when the claimed software is in the useful arts. Properly drawing this line, however, requires more than a tautological statement of the obvious. The analytic structure must be able to accurately test the true nature of each particular claim to a software invention for compliance. Only by ensuring that exclusively deserving software wines are put into patent bottles can the patent system lay claim to "appellation controle."

Measured against this decision-making backdrop the administratively appealing hardware focused approach, particularly as articulated in the Guidelines, fails to meet the need.¹³ The talismanic reliance on the presence or absence of computer readable carriers results simultaneously in "over" and "under" inclusive application of the patent incentive to software inventions.¹⁴

11. The Supreme Court has consistently focused on the "first principles," referring primarily to the Patent Act. See *Diamond v. Chakrabarty*, 447 U.S. 303, 307-309 (1980); *Diamond v. Diehr*, 450 U.S. 175, 182-84 (1981). The CAFC has followed this lead. See *In re Alappat*, 33 F.3d 1526, 1542-43 (Fed. Cir. 1994). To fully understand software inventions in the context of the statutory first principles requires one further step back to the underlying "promoting progress in the useful arts" policy objectives contained in the Patent Clause of the Constitution. See *infra* notes 34-40, 194-238 and accompanying text.

12. This assumes we approve of the current Constitutional objectives. Times change and there is no on-going guarantee that we have the patent incentive right in the Constitution or, even if we do, that Congress has been sufficiently circumspect in tailoring its use of the power to apply the patent incentive. One can, therefore, challenge whether the existing Constitutional and statutory objectives are appropriate to current social and economic conditions and, in particular, as they may be reflected in the software industry. In all events, unless and until a policy change occurs, we must ensure that the patent laws are at least operating in a fashion consistent with the existing policy objectives. See *infra* note 238 and accompanying text.

13. The *Guidelines* point out they are merely intended as a restatement of the current law as determined by the statute and the courts. *Guidelines, supra* note 4, at 7479. The problems are, therefore, in part the fault of the current regime created by the courts. However, the *Guidelines* also reflect PTO over-reaction in their interpretation of the CAFC "hardware" decisions which, although understandable, results in over- and under-inclusive treatment of software related inventions. See *infra* note 14.

14. See *infra* notes 248-277 and accompanying text. The "article of manufacture" approach under the *Guidelines* can be criticized as an exercise in "form over substance" drafting. As with any form over substance situation, manipulating the system results in distorting its operation when compared to the underlying objectives. See John R. Thomas, *Of Text, Technique, and the Tangible: Drafting Patent Claims Around Patent Rules*, 17 J. Marshall J. Computer & Info. L. 219 (1998).

Under the current state of play, over-inclusion is most apparent. Because the Guidelines overstate the case for article of manufacture "hardware carrier" style software claims as technology, a significant opportunity exists to smuggle non-useful arts claims into the patent system¹⁵ under their guise.¹⁶ In addition, the resulting failure to clearly and properly define the actual nature of software inventions by applying the patentable subject matter analysis leads to inadequate identification of prior art and insufficiently stringent review for novelty and non-obviousness.¹⁷

As a consequence, examination of software applications is inappropriately weakened resulting in a disruption of the economic cost-benefit balance underlying the U.S. patent system.¹⁸ Rather than providing appropriate incentives for innovation in the technological arts, the current system offers the possibility of obtaining patents covering inventions outside the targeted class. This results in an industry rush to build ever more impressively sized portfolios of at best marginal and at worst inappropriate patents couched in software terms.¹⁹ Large amounts of time and energy which could be otherwise invested are directed to obtaining, assessing, avoiding and litigating software patents which have come through an examination process that makes inadequate distinction between justified and unjustified exclusionary rights.²⁰ One must take seriously the specter of a software industry dominated by patent "haves"

15. Avoiding the smuggling of non-technological inventions such as laws of nature or unpatentable processes through the use of software has been a primary concern of the earlier approaches to the question of patentability of software. See *infra* notes 45, 76-86, 212-229 and accompanying text.

16. See *infra* notes 261-271 and accompanying text.

17. See *infra* note 54 and accompanying text.

18. See *infra* notes 34-40, 194-238 and accompanying text (discussing the policy objectives of the United States patent system). The problem with weakening of the examination requirements was most dramatically demonstrated by the complete elimination of the examination requirement in the 1793 Patent Act under which patents were granted to "inventors" patents based merely on a filing. As a consequence of the enormous number of frivolous assertions, the examination requirement was reinstated in 1836. See DONALD S. CHISUM, CHISUM ON PATENTS § OV-5 n.10 (1997) (citing the 1836 Senate Committee Report); GREGORY A. STOBBS, SOFTWARE PATENTS § 1.11 (1996).

19. The number of software patents issued has continued to increase significantly over the last 10 years. See, e.g., G. Aharonian, Internet Patent News Service, *1995 US Software Patent Statistics*, (visited Oct. 15, 1998) <<http://lpf.ai.mit.edu/Patents/1995-patents.html>> (an increase from approximately 2038 in 1987 to 6951 in 1995); D. Syrowik & R. Cole, *The Challenge of Software-Related Patents* (Software Patent Institute) (visited Oct. 15, 1998) <<http://www.spi.org/primintr.htm>>; Finnegan, Henderson Growing Pains: *Software Patents Come of Age* (visited Oct. 15, 1998) <<http://www.finnegan.com>> (citing an increase from 3,270 in 1987 to 11,714 in 1995). The increase may in part reflect a need to keep up with others by developing a "defensive" portfolio against future assertions.

20. Although little direct empirical evidence is available concerning corporate operating budgets, which are confidential, the growth in patent application filings and reported

holding each other at bay by brandishing large binders of patents as deterrent threats of mutual destruction while offering the “have nots” unattractive, or even impossible, “pay or perish” decisions. The inappropriate application of the patent law incentive becomes a source of economic inefficiency, imposing unjustified costs,²¹ impeding progress in the very fields it was designed to promote, distorting competition and reducing rather than enhancing the general social welfare.

We must be equally wary of the less apparent under-inclusion lurking in the current software as article of manufacture analytic framework. The Constitutional policy objective is not to restrict or eliminate patent protection for software related inventions. It is to ensure that the patent incentive is directed in ways that generate invention and progress in the technological arts, including, when appropriate, in the field of software development.²² The imposition of a hardware carrier requirement,²³ which is unnecessary to proper application of the patent laws to software inventions, prevents direct protection of software innovation *per se*. The effects are not inconsequential. Inventors are forced to mischaracterize software innovations in hardware incarnations in a “real world” technological environment that is heading in precisely the opposite direction by moving computer system functionality from hardware to software.²⁴ The system currently, therefore, inhibits the ability to describe and protect the true nature of software inventions under the patent laws. In addition, including the unnecessary carrier element in the claims provides a basis for irrelevant and unmerited non-infringement defenses completely unrelated to the underlying functionality (where the actual dispute resides) thereby unjustifiably increasing enforcement costs. The overall result is too little incentive to innovations in this important field of technological endeavor.

patent infringement lawsuits are indicative of a significant trend. *See supra* note 19 and accompanying text.

21. Even if many of these patents are ultimately invalidated, the costs associated with dealing with the resulting assertions can dramatically affect the cost-benefit scales underlying the patent system, with the disproportionate operating costs overwhelming the value of the innovations encouraged.

22. *See* the applicant’s unsuccessful argument in *In re Trovato*, 42 F.3d 1376, 1383 (Fed. Cir. 1994), *vac. and remanded*, 60 F.3d 807 (Fed. Cir. 1995) (en banc); *see infra* notes 272-277 and accompanying text.

23. Some article of manufacture inventions do relate to the medium itself and the way it stores the program or other information. These claims are to the hardware itself, independent of the content, and do not directly implicate the software as article of manufacture question. *See infra* note 240 and accompanying text.

24. There is a shift from hardware to software implementation as part of an overall convergence on digital technology in the field of computing, including its computational, video and communication aspects. *See, e.g., Mr. Fix-it: Interview with Tim Thorsteinson*, Portland Bus. J., Jan. 23, 1998; *see infra* note 277 and accompanying text.

III. ANALYSIS

This Article argues that the solution lies in distinguishing between two possible functional uses of software²⁵ when making the “useful arts” policy inquiry. First, software can act as the specific technological vehicle for implementing new or existing activities as part of a computer system, the actual instructional program for the hardware. Second, software can serve as a language, which like any other language is a symbolic vehicle for communicating an underlying idea, in this instance the contained methodology.

The test for whether a software invention constitutes patentable subject matter should be tied expressly to drawing this implementation versus a language dichotomy²⁶ based on the patent claims.²⁷ If the claims only cover the use of the software functionality as instruction in a computer system the software serves solely in its implementation role, then such claims should be routinely treated as patentable subject matter. In contrast, if the claims are not so limited then software terminology is being used as a language to express an idea. This situation does not make the functionality claimed *per se* unpatentable. It does require, however, that the claims be assessed based solely and directly on the underlying functionality, apart from the fact that software language has been used as the means for its expression.

By eliminating the faulty reliance on the presence or absence of a hardware carrier,²⁸ the implementation versus language test resolves both the over-inclusion and under-inclusion difficulties in the current article of manufacture regime. Over-inclusion disappears because claims, which reach beyond computer system implementation of the software functionality, cannot hide behind mere inclusion of computer readable media as a means to sneak through § 101. Under-inclusion is eliminated because the test permits direct treatment of “software as implementa-

25. See *infra* notes 243-247 and accompanying text.

26. See discussion *infra* Part III.G. The similar sounding “idea-expression” dichotomy has been used in the debate over the applicability of copyright to computer software, setting up a basic division between the “code” expression and the underlying idea. The former is protected under copyright law and the latter is not. This copyright model does not, however, capture the dichotomy proposed here for patent law purposes. Although the “idea” exclusion from copyright identifies the aspects of a software program which may be subject to patent law protection, a further distinction is required to assess software inventions for patentability under § 101. 35 U.S.C. § 101 (1994). That distinction is supplied by the proposed differentiation between software as language and software as technological implementation as part of a computer system. See *infra* note 285 and accompanying text.

27. See discussion *infra* notes 65-68 and accompanying text (discussing the role of the claims in patent law).

28. It is similarly irrelevant whether the computer system itself is included in the claims or specification. The analysis applies equally to “machine” claim formats. See *supra* note 10 and accompanying text.

tion" functionality as patentable subject matter without forcing such inventions into hardware formats or requiring surplus claim elements to avoid potential rejection under inappropriate process tests.

To be effective an appropriate patentable subject matter test must also permit and enable proper operation of the other patentability requirements. The proposed implementation versus language model does so. First, it clarifies how the *progress* of the useful arts objectives inherent in the § 102 novelty and § 103 nonobviousness reviews should apply to software claims. For software as implementation inventions the novelty and nonobviousness must be found in the technology which constitutes the substance of these claims: normally the computer technique or the fact of computerization (automation) itself.²⁹ On the other hand, software as language inventions are not permitted to look to the computer software label or expression as a source of novelty. They must stand or fall solely on the progress or innovation found in the underlying functionality expressed. Additionally, clarifying which type of software invention is under review, computer technology or underlying functionality, facilitates the location of, and comparison with, appropriate prior art by applicants, examiners and the courts. Finally, the model avoids any distorted use of the § 112 definiteness and enablement inquiry as an ineffective tool for distinguishing between software as implementation and software as language inventions based on abstract idea notions.³⁰

The proposed model will result in a more appropriate application of the patent incentive to the field of software technology. By avoiding the over-inclusion problem inherent in the present analytical framework, the proposed model helps eliminate undesirable software applications and patents. The software as implementation limitation under § 101 and the increased efficacy of the § 102 and § 103 reviews will likely result in less frequent grants of software related patents. The patents which actually do issue will be more clearly defined and of more focused scope. The net result will be fewer but more appropriate and stronger software patents, precisely the types of patents current patent policy strives to promote industries internalizing as the cost of providing appropriate incentive to software inventors.

A transition to the new model can be effected without significant effort. Because much of the existing software patentability muddle is of judicial origin, the courts can equally quickly retrieve us. The CAFC can adopt the proposed implementation-language analytic framework simply by modest clarification of the rationale supporting their current article of

29. See *infra* notes 320-338 and accompanying text.

30. See *infra* notes 229, 302-311 and accompanying text.

manufacture line of analysis.³¹ With this accomplished, the PTO can modify the Guidelines and the system will be fully operational.

This article develops the case for the proposed implementation versus language analytic framework in several steps. Part A briefly sets out the origins and basic objectives of the United States patent system and explains the roles played thus far by the primary United States patent law requirements in the software patentability debate. Part B provides background and context for the article of manufacture discussion. Specifically, Part B provides a concise historical review of the development of the software patentability debate from the original appellate decisions through the various mathematical algorithm formulations to the CAFC's *In re Alappat*³² and *State Street Bank & Trust v. Signature Financial Group*³³ "machine" decisions. Part C provides a discussion of the evolution of the current article of manufacture approach, including the relevant 1994 CAFC decisions and the PTO Guidelines. Parts D, E and F explain why the current article of manufacture approach set out in the Guidelines gives rise to the problems of over-inclusion and under-inclusion. Part G describes the implementation-language alternative system of analysis and explains why it consistently provides results congruent with the policy objectives of the United States patent system.

A. ORIGINS OF UNITED STATES PATENT LAW AND RELEVANT PATENT
LAW BASICS FROM WHENCE THEY CAME AND WHERE THEY
SEEK TO GO

The general origins of United States patent law are relatively clear. After a brief debate between James Madison and Thomas Jefferson concerning the general undesirability of monopolies; Jefferson relented in the case of incentives for certain types of inventive behavior.³⁴ The United States Constitution was drafted to include an express congressional power "to promote the progress of science and useful arts, by se-

31. See *infra* notes 254-258 and accompanying text. It would, of course, be appropriate at the next opportunity for the Supreme Court to expressly overrule *Gottschalk v. Benson*, 409 U.S. 63 (1972) and *Parker v. Flook*, 437 U.S. 584 (1978) and eliminate the useless mathematical algorithm framework. See *infra* note 85 and accompanying text. Even though this framework can be avoided under the proposed article of manufacture approach, it is undesirable to have it hanging around like a guest overstaying his welcome in a spare room. One can never tell when it may emerge to cause mischief.

32. *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994).

33. *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368 (Fed. Cir. 1998).

34. Because of his general opposition to monopolies, based on the English experience, Jefferson originally proposed an "anti-monopoly" provision in the Constitution. He eventually was persuaded by Madison that the special incentive case supported a limited monopoly for patents and copyrights. There is an excellent review of the Constitutional origins in *Graham v. John Deere Co.*, 383 U. S. 1, 5-11 (1966). See STOBBS, *supra* note 18, § 1.7; 1 EARNEST B. LIPSCOMB III, WALKER ON PATENTS § 2.1 (3d ed. 1984).

curing for limited times to authors and inventors the exclusive right to their respective writings and discoveries."³⁵ This provision not only authorized the creation of a federal patent law but, uniquely among the grants of congressional power, also expressly set out the objective and even some details concerning the means. An incentive in the form of a limited right to preclude competition could be offered to inventors to encourage the investment of their time and resources in inventive endeavors which would progress the useful arts. The hope was that this encouragement would result in the desired growth in the useful arts which, in turn, would result in overall economic growth and a general increase in prosperity.³⁶

The Constitutional provision does not require Congress to create such incentives, it merely empowers it to do so.³⁷ At the founding of the nation pro-industrial sentiment was strong and Congress acted with alacrity, adopting the first United States Patent Act in 1790.³⁸ The United States has had a patent law ever since. Although there have been numerous revisions,³⁹ there has been virtually no change in the patent law's generalized pursuit of the Framers' desire to incentivize progress in the useful arts.⁴⁰ This objective remains the driving force

35. U.S. CONST., art I, § 8, cl. 8. This provision authorizes both the federal patent and copyright laws. See *infra* note 201 and accompanying text (discussing the respective authorizations).

36. The United States patent law is predicated on the view that society has an economic interest in incentivizing inventive behavior, not that the inventor has a natural or moral right to possess what she invents. *Diamond v. Chakrabarty*, 447 U.S. 303, 308 (1980); *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 480 (1974); *Graham*, 383 U.S. at 8-9; see generally President's Commission on the Patent System, 2 Weekly Comp. Pres. Doc. 1752 (Dec. 2, 1966). In addition to granting an exclusive right as an economic incentive, the U.S. patent laws require full disclosure of the invention to ensure it is available for public use at the end of the patent term. See *infra* notes 65-68 and accompanying text (discussing 35 U.S.C. § 112 (1994)).

37. Like the other provisions of section 8 of article I of the Constitution, the patent/copyright clause is a grant of congressional power, not a requirement.

38. See Chisum, *supra* note 18, § OV-2; STOBBS, *supra* note 18, § 1.6.

39. There have been at least five rewrites of the U.S. patent laws, with the last major revision adopted in 1952. In addition, there have been a variety of amendments. See Chisum, *supra* note 18, Overview § OV-9-15; STOBBS, *supra* note 18, §§ 1.6-1.12.

40. Despite the numerous changes many of the fundamentals have remained consistent since the earliest laws. See STOBBS, *supra* note 18, § 1.13 ("Although the law has been amended on numerous occasions—and even rewritten twice since 1836—no basic changes have been made in its general character in the succeeding one hundred and thirty years" (quoting President's Commission on the Patent System, 2 WEEKLY COMP. PRES. DOC. 1752 (Dec. 2, 1966)). Many amendments have been articulated as simply clarifications of existing practice, with the Supreme Court finding that earlier decisions continue to apply under the new statute. See, e.g., *Graham*, 383 U.S. at 17 (the addition of § 103 in 1952 was a codification of earlier judicial decisions); *Dann v. Johnston*, 425 U.S. 219, 226 (1976) (to the same effect); *Diamond v. Diehr*, 450 U.S. 175, 182-84 (1981) (the change of the word "art" to "process" in § 101 in 1952 was a linguistic updating, not a significant substantive

behind the patent system and, therefore, continues to provide important guidance when determining to what extent claims to software related inventions should receive patent protection.

1. *The Basic Patentability Inquiry*

The Patent Act and associated PTO regulations and guidelines contain a number of requirements which must be met to obtain a patent.⁴¹ The five key requirements for purposes of the present inquiry include the patentable subject matter restriction, novelty, nonobviousness, usefulness and adequate description (identification and enablement). Each of these requirements plays a particular role in ensuring that the patent incentive furthers the desired Constitutional objective of promoting progress in the useful arts for the general public's benefit.

The patentable subject matter requirement keeps the patent incentive aimed exclusively on endeavors within the Constitutionally targeted useful arts. Section 101 of the Patent Act⁴² lists the following classes of patentable subject matter: "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof."⁴³ The breadth and generality of this language,⁴⁴ when coupled with the two different functions of computer software described above, has made actual application somewhat problematic. Much of the software as patentable subject matter debate reflects a concern that

change). In contrast, there have been numerous ebbs and flows in the judiciary's disposition toward patents. See CHISUM, *supra* note 18, § OV-9-15. See also discussion *infra* notes 194-238 and accompanying text.

41. The basic patenting process involves filing an application with the PTO. The application is examined by a PTO examiner to determine whether the requirements have been met, including those under §§ 101, 102, 103 and 112 discussed below in this section of the article. See Chisum, *supra* note 18, Overview at OV-1-2.

42. 35 U.S.C. § 101 (1994).

43. *Id.* Except for the definition of "process" in 35 U.S.C. § 100(b) (1994), none of these terms are defined in the statute. See *In re Chatfield*, 545 F.2d 152, 159 (C.C.P.A. 1976), *cert. denied*, 434 U.S. 875 (1977) (Rich, J., dissenting) (reviewing the evolution of the language of 35 U.S.C. § 101). It does not matter for purposes of the statutory subject matter requirement which in particular of the four categories covers the invention. See *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1372 (Fed. Cir. 1998) ("We note that, for the purposes of a § 101 analysis, it is of little relevance whether [a claim] is directed to a 'machine' or a 'process,' as long as it falls within at least one of the four enumerated categories of patentable subject matter . . ."), *Id.* at 1372 n.2 "The first door which must be opened on the difficult path to patentability is § 101 . . . If the invention . . . falls into any one of the named categories, he is allowed to pass through to the second door, which is § 102 . . ." *Id.* (citing *In re Bergy*, 569 F.2d 952 (C.C.P.A. 1979)). It can, however, matter to the patentee which category covers the invention for enforcement purposes. See discussion *infra* notes 92-94 and accompanying text.

44. Of particular note is the frequently cited statement from the Congressional history that the patent laws were meant to cover "anything under the sun made by man." See discussion *infra* notes 205-207 and accompanying text.

claims implicating unpatentable subject matter will be smuggled past the § 101 border guards by expressing them in software "technology" terms.⁴⁵ The patentable subject matter requirement has been, and continues to be, a prime focus of the software patentability debate.⁴⁶

The remaining requirements play additional roles in the patent policy framework. The § 102 novelty⁴⁷ and § 103 non-obviousness⁴⁸ "inventiveness" requirements and the § 101 usefulness⁴⁹ requirement all seek to ensure that the invention is truly an inventive step forward (progress) in the useful arts. Finally, to ensure the invention involves actual implementable technology, to give notice of the extent of the exclusive patent privilege and to make the invention available to society at the end of the

45. This is the genesis of the mathematical algorithm test developed by the Court in *Gottschalk v. Benson*, 409 U.S. 63 (1972), in an attempt to identify software claims which involved no more than claims to "laws of nature." See discussion *infra* notes 76-83, 212-227 and accompanying text. This same concern is reflected in the development and application of the various permutations of the FWA test which also reflect concern with overly broad claiming. See, e.g., *In re Trovato*, 42 F.3d 1376, 1382-83 (Fed. Cir. 1994), *vac. and remanded*, 60 F.3d 807 (Fed. Cir. 1995) (en banc); *In re Schrader*, 22 F.3d 290, 295 (Fed. Cir. 1994); see discussion *infra* notes 85-88 and accompanying text. Of the four categories of § 101 statutory subject matter, only composition of matter remains unimplicated. Because of its dual language and implementation roles, computer software claims have been variously described as apparatus (machine or article of manufacture) and process.

46. See, e.g., *Gottschalk v. Benson*, 409 U.S. 63 (1972); *Parker v. Flook*, 437 U.S. 584 (1978); *Diamond v. Diehr*, 450 U.S. 175 (1981) (discussing primarily 35 U.S.C. § 101 issues). Historically, the vast majority of CAFC/CCPA software cases involved only § 101. With the PTO's shift in *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994), to an article of manufacture and printed matter rejection, § 102 and § 103 became critical. The new *Guidelines* practice also seems to make clearing the § 101 hurdle primarily a drafting exercise with the emphasis on the § 102 and § 103 inquiries.

47. 35 U.S.C. § 102(a) (1994). Novelty addresses whether the claimed invention is "new." At one level this inquiry involves questions of inventorship including existing art and priority among inventors. At another level there is a question of whether discovery of a pre-existing law of nature or natural phenomena is really invention. The former is quite clearly addressed by the language of § 102. The latter is more difficult because the courts address it in both "statutory subject matter" and "novelty" terms. See discussion *infra* notes 59-64, 335 and accompanying text.

48. 35 U.S.C. § 103(a) (1994). The section reads in part

"if the difference between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains."

Nonobviousness extends the novelty inquiry to ensure that a claimed invention which may not have been specifically articulated previously is more than an obvious extension of prior knowledge to someone "of ordinary skill in the applicable art." See *Dann v. Johnston*, 425 U.S. 219, 225-26 (1976); *Graham v. John Deere Co.*, 383 U.S. 1, 13-19 (1966) (setting out the basic test).

49. 35 U.S.C. § 101 (1994). The requirement comes from the "new and useful" preamble to the enumeration of the statutory classifications. See *infra* note 69 and accompanying text.

patent term, § 112⁵⁰ requires a sufficiently detailed description to permit those skilled in the relevant field to clearly identify, understand and practice the invention.

The novelty and nonobviousness "inventiveness" requirements have had three important connections with the software patentability debate. First, they have been a matter of significant logistical concern to the PTO. Particularly at issue are whether adequate numbers of properly trained patent examiners can be employed and whether the technology that already exists (the "prior art") can be reliably identified, both being essential to properly determining if the applicant's invention is truly novel and nonobvious. These pragmatic institutional concerns have been one motivation for the PTO's long standing position that computer software was not patentable subject matter.⁵¹ Although institutional considerations should not be used indirectly to undermine patent system objectives, we must recognize that for the patent system to operate properly, appropriate resources are required. To date the issues have been managed, albeit with some difficulty.⁵² However, if it becomes clear proper resources do not exist or are not funded, then the policy objectives need to be re-thought in light of those realities.⁵³

Second, because the confused state of the patentable subject matter debate under § 101 frequently prevents a clear determination of the precise nature of the invention at issue, many software related inventions fail to have any significant applicable prior art cited against them.⁵⁴ This is an inappropriate and unfortunate result. The patentable subject matter question is only one of the patent policy gatekeepers. If the pat-

50. 35 U.S.C. § 112 (1994).

51. In effect, if computer software is not invited to the patent party at all there is no need to deal with the logistical problems surrounding novelty or nonobviousness. These PTO concerns influenced the early Supreme Court decision in *Benson*. See *Gottschalk v. Benson*, 409 U.S. 63, 72 (1972); *Parker v. Flook*, 437 U.S. 584, 587-88 (1978). See also *Diehr*, 450 U.S. at 218 (Stevens, J., dissenting).

52. The examiner issue is addressed by PTO recruiting efforts, made difficult by the fact that software engineers are in high demand in industry giving them attractive alternatives to government work. The prior art situation is being actively addressed in a number of ways, including private industry efforts. See, e.g., R. Stern, *On Defining the Concept of Infringement of Intellectual Property Rights in Algorithms and other Abstract Computer-Related Ideas*, 23 AIPLA Q. J. 401, 408-409 n.17 (1995).

53. The wisdom of recent diversions of significant amounts of PTO revenues by Congress should be assessed in this light.

54. The CAFC cases demonstrate the point. The vast majority involve only § 101 issues. The parties exhaust themselves on this issue with examination as a practical matter based exclusively on the outcome of that debate. For those few cases which do raise § 102 or § 103 questions, once the CAFC clears the § 101 problem by reclassifying the invention, frequently the original § 102 or § 103 references disappear in view of the inapplicability of the cited prior art to the new view of the invention. See, e.g., *In re Lowry*, 32 F.3d 1579, 1584 (Fed. Cir. 1994).

ent system is to operate properly the § 101 review must adequately inform the novelty and nonobviousness examination to ensure those requirements are also satisfied. This is particularly true in the case of many computer software inventions, where the key question frequently is whether the claimed computer implementation is obvious given the current state of computer technology.

Third, the inventiveness requirements have in the past become entangled with the § 101 analysis, confusing both discussions. At a policy level the two tests embody separate objectives. The patentable subject matter requirement ensures that the incentive is being applied to activities in the proper field of human endeavor, those in the "useful arts." The inventiveness requirements address whether the particular results represent advances or progress in those arts. Finding that an invention involves patentable subject matter says nothing about the "inventiveness" aspects of novelty and non-obviousness or vice-versa.⁵⁵ These separate issues should involve separate inquiries.⁵⁶

A pair of simple examples can demonstrate why this is. Dramatic works are not technology and do not come within the "useful arts" target class of inventions entitled to the patent incentive. As an entire category they simply are not patentable subject matter.⁵⁷ This determination is entirely independent of questions of novelty. Non-patentable subject matter remains non-patentable subject matter no matter how novel it may be. It is irrelevant whether the particular play is new and nonobvious or a verbatim copy of a previous work.

Conversely, lack of novelty in no way negates the fact that a particular invention is § 101 patentable subject matter. For example, a patent application covering a computer dynamic random access memory chip clearly involves patentable subject matter under § 101. If the same in-

55. See *Diehr*, 450 U.S. at 189-190

("Section 101 . . . is a general statement of the type of subject matter that is eligible for patent protection Specific conditions for patentability follow and § 102 covers in detail the conditions relating to novelty. The question. . . of whether a particular invention is novel is 'wholly apart from whether the invention falls into a category of statutory subject matter.'");

State Street Bank & Trust v. Signature Fin. Group, 149 F.3d 1368, 1372 n.2 (Fed. Cir. 1998) ("The first door which must be opened on the difficult path to patentability is § 101 If the invention . . . falls into any one of the named categories, he is allowed to pass through to the second door, which is § 102 Notwithstanding the words 'new and useful' in § 101, the invention is not examined under that statute for novelty") (citing *In re Bergy*, 569 F.2d 952 (C.C.P.A. 1979).

56. See *Diehr*, 450 U.S. at 191; *In re Warmerdam*, 33 F.3d 1354, 1360 (Fed. Cir. 1994) (merely because patentable subject matter involved does not mean a patent will issue, the § 102, § 103 analysis must also be performed).

57. Although unlikely, something in the play's contents may be patentable. For example it may contain a novel, non-obvious process for performing a particular task. Even in this case, however, it is the process, not the play as a play which is patentable.

vention has been previously patented, no patent will issue because the previous patent demonstrates lack of novelty. This finding of lack of novelty, however, does not make the memory chip any less patentable subject matter, any more than a finding of novelty makes the play § 101 subject matter.⁵⁸

Despite this lack of inter-relationship, the Supreme Court's *Parker v. Flook*⁵⁹ decision in 1978 appeared to legitimize a "point of novelty" approach to the § 101 inquiry. The Court's approach involved separating the claimed invention into old and new aspects as a way of identifying where the invention (the "new") was located and then testing only these "inventive" aspects under § 101.⁶⁰ This technique quickly became popular with the PTO in the early days of the software as mathematical algorithm test, as it permitted the examiner to parse the claims, identify the invention as the underlying unpatentable mathematical formula and thereby justify a § 101 rejection.⁶¹ A three-year debate followed over whether a search for the "point of novelty" in the invention was an appropriate part of the patentable subject matter inquiry.

The approach was fully and finally discredited by the Supreme Court in *Diamond v. Diehr*.⁶² The majority opinion in that case pointed out that under a "point of novelty" parsing exercise, any invention in which the novelty involved a combination of old elements or old and new

58. See *Diehr*, 450 U.S. at 191 ("A rejection on either [102 or 103] grounds does not affect the determination that respondents' claims recited subject matter which was eligible for patent protection under § 101.").

59. *Parker v. Flook*, 437 U.S. 584 (1978).

60. The majority opinion in *Flook* was slightly indirect, permitting it to pay at least lip service to the "invention as a whole" requirement. *Flook*, 437 U.S. at 593-94. The Court "assumed" the included mathematical algorithm was in the prior art (was old) and then analyzed only the remaining material, not surprisingly finding no invention. *Id.* But by finding there was no invention, the Court was addressing novelty, not patentable subject matter. Justice Stevens explained his *Flook* opinion in his dissent in *Diehr* by indicating it rests on the distinction between "discovery" in § 101 and "novelty" in § 102. *Diehr*, 450 U.S. at 211 (Stevens, J., dissenting). The "discovery" concept, meaning finding things which previously exist, however, has two aspects. Under § 101, merely claiming a newly discovered, but pre-existing, law of nature standing alone does not claim a useful art, and therefore, is not patentable subject matter. See discussion *infra* notes 214-222 and accompanying text (laws of nature not within the "useful arts"). However, when such discoveries are applied to a specific technological solution making the invention as a whole patentable subject matter, the question remains whether the discovery of the law of nature (or other non-technological subject matter) can form the basis for a § 102 novelty determination. This article argues that because novelty considerations are limited to "progress of the useful arts," such a discovery, standing alone, cannot form the basis for novelty. Although Justice Stevens' argument has considerable merit in the novelty context, it is not appropriate in the statutory subject matter context. See discussion *infra* notes 330-335 and accompanying text.

61. See *Diehr*, 450 U.S. at 179, 189 n.12 (noting this was the PTO's position).

62. *Id.* at 187-89.

elements would be improperly rejected.⁶³ Therefore, the Court held that the "claims must be considered as a whole," without consideration of novelty, for purposes of § 101.⁶⁴

The description and enablement requirements under § 112 play a central role in any patentability inquiry.⁶⁵ The most critical portion of any patent application is the specification, consisting of the detailed description and the claims, which together define and limit the actual invention for which the patent is sought.⁶⁶ The specification is, therefore,

63. *Id.*

64. *Id.* In particular, the Court rejected the parsing of a claim involving a law of nature or mathematical formula under § 101, as the fundamental inquiry was whether the invention *applied* the admittedly unpatentable subject matter to create a patentable invention when viewed as a whole. *Id.* at 192. See discussion *infra* notes 226-227. The CAFC has routinely embraced the "invention as a whole" approach to the § 101 inquiry. See, e.g., *Arrhythmia Research Tech. v. Corazonix*, 958 F.2d 1053, 1057 (Fed. Cir. 1992); *In re Meyer*, 688 F.2d 789, 796 (C.C.P.A. 1982). The PTO also fully adheres to the "claims as a whole" analysis. See discussion *infra* notes 167-168 and accompanying text. Although it is inapplicable in the context of the § 101 statutory subject matter inquiry, the parsing exercise, for the very reasons articulated by Justice Stevens, is extremely relevant to the novelty inquiry itself. See discussion *supra* note 60 and accompanying text.

65. 35 U.S.C. § 112 (1994) ("setting" out various requirements). These include that the invention be fully described and circumscribed (detailed description under paragraph one of § 112 and particularly point out and distinctly claim under paragraph two of § 112) and that the specification as a whole enables a person skilled in the art to make and use the invention without undue experimentation (enablement and best mode under paragraph one of § 112). This "definitional" role of § 112 is important in three respects beyond its § 101 role described in the text. First, a clear definition gives notice of the extent of the exclusionary patent privilege. Second, the enablement requirement ensures the invention involves actual implementable technology. See discussion *infra* notes 305-311 and accompanying text. Third, the publication of this information provides dissemination to society for general use following expiration of the patent term.

66. The claims define the scope of the patent exclusionary rights. If the patent is granted, only matters which "read on" (meaning are covered by) the language of the final claims, as elaborated by the detailed description or under the doctrine of equivalents, will infringe. See 35 U.S.C. § 271 (1994); *Corning Glass Works. v. Sumitomo Electric U.S.A.*, 868 F.2d 1251, 1257-58 (Fed. Cir. 1989) (basic infringement test); *Warner-Jenkinson Co. v. Hilton Davis Chemical Co.*, 520 U.S. 17 (1997) (doctrine of equivalents).

Complicating "scope determination" of many software patent claims is paragraph 6 of § 112, which authorizes "means for" claiming as follows:

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

35 U.S.C. § 112 (1994).

As the language indicates, claims in this form do not state the specific means for performing the function. Instead the detailed description provides context for and amplifies the general "means for" elements of the claim. This permits an applicant to expand the scope of the claims efficiently. For example, rather than limiting a claimed computer software article of manufacture invention by using "a diskette containing a software program" a "means for" claim might read as "a computer readable means for storing a software

the basis on which a software related invention should be characterized as an "article of manufacture," a "process"⁶⁷ or something else. The claims, in particular, must be the focal point for any framework used to determine whether an application describes and is limited to § 101 patentable subject matter.⁶⁸

The last requirement, "usefulness"⁶⁹ historically has been called into service to justify arcane results like the non-patentability of perpetual motion and gambling machines. The usefulness hurdle currently is extremely low, effectively inquiring whether the invention performs as

program" and the related detailed description might indicate computer readable means "includes a diskette, ROM, DRAM or other computer readable memory devices as known in the art." This format has given rise to a prolonged debate concerning the scope and limitations of this type of claim under the "equivalents thereof" language in § 112, paragraph 6. The current trend is in the direction of construing the language narrowly, limiting it to those items specifically mentioned and their true ("literal") equivalents. See *In re Donaldson*, 16 F.3d 1189 (Fed. Cir. 1994) (en banc). It remains clear, however, that read broadly or narrowly the resulting claim scope still acts as a limitation on the claim. Failure to include some specific structure in the detailed description accompanying in a means for claim will result in an indefiniteness rejection under § 112.

67. Using "means for" claim structure in computer software cases frequently results in claims which simply track the logic (algorithm) of the software program, *i.e.*, a listing of the steps to be performed in a particular order, rather than the specific software/hardware implementation. See discussion *supra* note 66 and accompanying text. Even if not claimed expressly as method or process inventions software claims in this format look like process claims. As a consequence the examiner may apply the process tests rather than the hardware tests under § 101. See discussion *infra* notes 177-183, 269-271 and accompanying text. Particularly under the "mathematical algorithm" approach this makes a rejection substantially more likely. The new "hardware" formats are an attempt to avoid this treatment of software claims as process inventions in examination. See discussion *infra* note 94 and accompanying text.

68. Section 101 cases will commonly "summarize" the invention. This can be helpful in understanding the general nature and objectives of an invention. Because the summary may actually reflect the judge's conclusion on the statutory subject matter issue it can, however, be a seriously misleading substitute for an actual § 101 claims analysis. See, *e.g.*, *In re Alappat*, 33 F.3d 1526, 1541, 1564 (Fed. Cir. 1994) (setting out the differing views of the majority and the dissent concerning the invention at issue). It remains, ultimately, the language of the claims which govern the scope of the patent and the related right to exclude others. It is, therefore, only the claims, read in light of the rest of the specification, which must be analyzed to make the patentable subject matter determination. The § 112 definiteness requirement has also occasionally been confused with the § 101 "abstract idea" requirement in software patent analysis. They are different concerns and should be kept apart. See discussion *infra* notes 229, 302-311 and accompanying text.

69. Although it echoes the Constitutional "useful arts" terminology this requirement arises from the "new and useful" language of 35 U.S.C. § 101 (1994). See *Brenner v. Manson*, 383 U.S. 519, 529 (1966). Although the language appears in § 101, the "usefulness" test is distinct from the basic statutory subject matter inquiry which derives directly from the "useful arts" limitation in the Constitution.

claimed regardless of the purpose.⁷⁰ To date, the requirement has not had a significant effect on the software patentability debate.

The Guidelines employ a form of "usefulness" test (termed "utility"). Although the terminology sounds similar, the Guidelines "usefulness" test is actually related to the patentable subject matter requirement: acting as an examiner's guide to whether an invention has a practical application with "real world" value in the technological arts as opposed to being a mere idea or concept.⁷¹ Although the Guidelines contain some serious implementation difficulties,⁷² this article argues that the PTO was certainly on the right road, or at least intersected with the right road, in moving in this direction. Focusing on "usefulness/utility" of a software invention in a "useful arts" sense contains the key to resolving the software as patentable subject matter conundrum.

B. A SHORT HISTORY OF SOFTWARE PATENTABILITY UP TO ARTICLE OF MANUFACTURE CLAIMS: HOW THE WORLD STARTED TO CHANGE WITH *Alappat*

The first significant software case, *In re Prater*,⁷³ was decided by the

70. There are a variety of views of "utility." One form of inquiry is whether invention does anything to any purpose. A more focused inquiry is whether the invention actually works to perform the function ascribed to it. See *Newman v. Quigg*, 877 F.2d 1575, 1581 (Fed. Cir. 1989), *cert. denied*, 495 U.S. 932 (1990) (a "perpetual motion machine" denied patentability for lack of usefulness). A final approach to "usefulness" involves more difficult line drawing concerning whether the "immorality" of an invention makes it undeserving of patent protection. See Chisum, *supra* note 18, §§ 4.01-4.04; ROBERT P. MERGES ET AL, *INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE* 163-166 (1st ed. 1997).

71. See *Guidelines*, *supra* note 4, at 7479. The *Guidelines* directly connect this concept of "utility" with the patentable subject matter test stating "the utility of an invention must be within the technological arts." *Id.* The *Guideline's* thus seem to contemplate using an additional usefulness test, distinct from the existing notion of usefulness, to determine whether an invention is merely an "abstract idea" and thus outside the "useful arts." The *Guidelines* clearly are directly on track to understanding the necessity of tying the useful arts limitation to the determination of when a software invention is an article of manufacture. See *infra* note 166 and accompanying text. The difficulty with the *Guidelines'* articulation of the usefulness test is that it can lead to confusion between testing the nature of the claimed invention itself and what the invention does against the useful arts requirement. See *infra* note 229 and accompanying text. In all events, rather than couching the issue in terms of utility which merely confuses this concept with the existing but irrelevant concept of usefulness, the PTO would have been better advised to address the point directly in "useful arts" terms. *Id.*

72. See discussion *infra* Part III.E.

73. See *In re Prater and Wei*, 415 F.2d 1393 (C.C.P.A. 1969). In *Prater*, the CCPA first articulated as dicta what was to become the 1994 hardware rallying cry:

No reason is now apparent to us why, based on the Constitution, statute, or case law, apparatus and process claims broad enough to encompass the operation of a programmed general-purpose digital computer are necessarily unpatentable. In one sense, a general-purpose digital computer may be regarded as but a storeroom of parts and/or electrical components. But once a program has been introduced,

Court of Customs and Patent Appeals ("CCPA"), the predecessor to the CAFC,⁷⁴ in 1968. In this and other early cases, the CCPA exhibited a generally robust view of the patentability of inventions arising out of the emerging computer software industry, eliminating the problematic "mental steps" doctrine and fairly consistently facing down a reluctant PTO.⁷⁵

This generally positive approach was abruptly cut off in 1972 by the Supreme Court's first word on the subject in *Gottshalk v. Benson*.⁷⁶ In *Benson* the Court announced in fairly incoherent terms⁷⁷ a § 101 test which precluded patent grants to any software invention which preempted the use of a mathematical algorithm.⁷⁸ The opinion, which can at best be described as skeptical concerning the patentability of software,⁷⁹ was interpreted by some (including most notably the PTO) to indicate that computer software inventions should be viewed as *per se* unpatentable subject matter.⁸⁰ The decision refocused subsequent debate between the CCPA, which continued generally to favor patentability, and the PTO which did not, on the meaning and extent of the Court's mathematical algorithm preemption test.⁸¹

the general-purpose digital computer becomes a special-purpose digital computer (i.e., a specific electrical circuit with or without electro-mechanical components) which, along with the process by which it operates, may be patented subject, of course, to the requirements of novelty, utility, and non-obviousness. Based on the present law, we see no other reasonable conclusion.

Id. at 1403 n.29.

74. The Court of Appeals for the Federal Circuit took over jurisdiction from the Court of Customs and Patent Appeals ("CCPA") on October 1, 1982, of all appeals from the Federal District Courts relating to patent matters as well as most appeals from PTO decisions. Federal Courts Improvement Act of 1982, Pub. L. No. 97-164, 96 Stat. 25. Before creation of the CAFC, appeals from District Court decisions went to the respective Courts of Appeals and appeals from PTO decisions to the CCPA. The CAFC has expressly adopted the prior decisions of the CCPA. See *South Corp. v. United States*, 690 F.2d 1368, 1370 (Fed. Cir. 1982).

75. See e.g., *In re Bernhart*, 417 F.2d 1395 (C.C.P.A. 1969); *In re Musgrave*, 431 F.2d 882 (C.C.P.A. 1970); *In re Waldbaum*, 457 F.2d 997 (C.C.P.A. 1972).

76. *Gottshalk v. Benson*, 409 U.S. 63 (1972).

77. See generally Chisum, *supra* note 8 (Professor Chisum analyzes the opinion virtually paragraph by paragraph demonstrating its numerous difficulties).

78. *Benson*, 409 U.S. at 72.

79. *Id.* at 71 ("It is said that the decision precludes a patent for any program servicing a computer. We do not so hold."); *id.* at 72 ("It may be that the patent laws should be extended to cover these programs . . .").

80. See, e.g., *In re Freeman*, 573 F.2d 1237, 1244 (C.C.P.A. 1978); *In re Chatfield*, 545 F.2d 152, 155-56 (C.C.P.A. 1976), *cert. denied*, 434 U.S. 875 (1977).

81. The more cynical would also say the *Benson* decision also launched a long and unnecessary period of finding appropriate limitations to what was a relatively unsophisticated, and in many ways erroneous, approach to the question of software patentability. Arguably, the development of the *Freeman-Walter-Abele* two step-test is more about limiting the unfortunate effects of *Benson* and *Flook* than the development of a rational ap-

The CCPA-PTO contest was further complicated by the Supreme Court's 1978 decision in *Flook* which reinforced the *Benson* mathematical algorithm preemption approach. The decision not only strengthened the view that the Court believed the entire class of computer software related inventions was potentially unpatentable subject matter, but added the confusion of the "point of novelty" approach to the analysis discussed above.⁸² The PTO embraced the decision, setting off yet another round of § 101 rejections and appeals to the CCPA.⁸³

The rising tide against the patentability of software related inventions finally started to recede with the Supreme Court's pivotal 1981 decision in *Diehr*. Following *Benson*, the CCPA had begun to develop what was to become the *Freeman-Walter-Abele* ("FWA") two-step test⁸⁴ as a vehicle for controlling the confusion caused by the mathematical algorithm approach.⁸⁵ Although the Court did not expressly overrule *Benson* or *Flook*, or confirm the FWA approach in *Diehr*, the opinion discredited the argument that software was *per se* unpatentable and lent considerable support to the substantial limitations that the CCPA had

proach to the patentability of computer software. See discussion *infra* notes 84-88 and accompanying text.

82. See *Parker v. Flook*, 437 U.S. 584, 595-96 (1978); see discussion *supra* notes 59-64 and accompanying text (discussing the point of novelty approach).

83. See, e.g., *In re Bradley*, 600 F.2d 807 (C.C.P.A. 1979), *aff'd by equally divided court, sub nom. Diamond v. Bradley*, 450 U.S. 381 (1981) (Burger, C.J., not participating); *In re Phillips*, 608 F.2d 879 (C.C.P.A. 1979); *In re Walter*, 618 F.2d 758 (C.C.P.A. 1980).

84. Named after the three CCPA cases in which it was developed: *In re Freeman*, 573 F.2d 1237 (C.C.P.A. 1978); *In re Walter*, 618 F.2d 758 (C.C.P.A. 1980) and; *In re Abele*, 684 F.2d 902 (C.C.P.A. 1982). The basic test was outlined in *Freeman* and modified in *Walter* and *Abele*.

85. See *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1373-74 (Fed. Cir. 1998) ("The Freeman-Walter-Abele test was designed by the Court of Customs and Patent Appeals, and subsequently adopted by this court, to extract and identify unpatentable mathematical algorithms in the aftermath of *Benson* and *Flook*."). The mathematical algorithm approach caused fundamental confusion because of its uncomfortable fit with the algorithmic operation of a computer program and the mathematical basis for the operation of a digital computer. There is simply no clear or useful way to define the mathematical algorithm concept in the software context. See *infra* note 90 and accompanying text. Although software is designed to implement an algorithm, computer algorithms have no direct correlation with unpatentable laws of nature including pure mathematical relationships. Some computer algorithms are laws of nature, others are not. Saying software operates using an algorithm, therefore, confuses rather than advances the § 101 determination. Additionally, a computer operating using a mathematical "radix" (binary mathematics and logic operations) has no correlation with whether it is implementing a mathematical algorithm in the law of nature/pure mathematics sense. This confuses what the program is doing with how the computer operates. See *Bradley*, 600 F.2d at 811-12. Most of the FWA test is an effort to undo this confusion, but being bound by the *Benson* and *Flook* decisions cannot entirely escape the flaws in those decisions.

been building around those decisions through the FWA test.⁸⁶

In its final post-*Diehr* incarnation, the FWA test effectively narrowed the *Benson/Flook* mathematical algorithm § 101 rejection from virtually all software claims to software related inventions which involved a “mathematical algorithm” and no external physical connections or effects (read generously in favor of the applicant).⁸⁷ With the patentable subject matter inquiry directed by the FWA test, relative peace reigned between the CCPA/CAFC and the PTO concerning software patents through the 1980s and early 1990s.⁸⁸

In 1994 the software as patentable subject matter dispute between the CAFC and the PTO re-erupted on a new front. The conflict arose over the proper analysis of new hardware-software combination claim formats⁸⁹ which had been developed by applicants to address continuing difficulties under the FWA regime.

Because *Diehr* failed to expressly overrule either *Benson* or *Flook*, the FWA test remained predicated on, and limited by, the mathematical algorithm approach. Determining when the PTO or the CCPA might find claims involved a mathematical algorithm, meaning the application would have to run the FWA gauntlet, was something of a Delphic enterprise.⁹⁰ In addition, claims to software applications in which a mathematical model was a prominent feature and there was no demonstrable,

86. Although the Court does not cite to either *Freeman* or *Walter*, both of which had already been decided, the Court clearly indicated that *Benson* and *Flook* were limited to claims to unapplied mathematical algorithms, which the Court equated to “laws of nature” not computer programs. *Diamond v. Diehr*, 450 U.S. 175, 185 (1981) (“Our recent holdings in *Gottschalk v. Benson* and *Parker v. Flook*, both of which are computer-related, stand for no more than these long standing principles [referring to non-patentability of laws of nature without application.]”). The decision had the effect of finally derailing the PTO’s general unpatentability of software line of argument and limiting the effects of the mathematical algorithm confusion. See *supra* note 85 and accompanying text. However, issues remained. See discussion *infra* notes 90-91 and accompanying text.

87. See *Abele*, 684 F.2d at 907 (“applied in any manner to physical elements or process steps”).

88. There is a notable lack of cases appealing § 101 rejections to the CAFC during the period, with only three reported decisions between October 1982 and April 1994: *In re Grams*, 888 F.2d 835, (Fed. Cir. 1989); *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989); and *Arrhythmia Research Tech., Inc., v. Corazonix Corp.*, 958 F.2d 1053 (Fed. Cir. 1992). See McCutchen et. al, *The Future of Software Patents*, (visited Oct. 15, 1998) <http://www.mccutchen.com/ip/ip_2101.htm>. This did not mean that all issues had been resolved. See discussion *infra* notes 90-91 and accompanying text.

89. A sign of the intensity of the discussion was that during 1994, the CAFC dealt with five appeals of PTO decisions relating, more or less directly, to these new claim formats.

90. Neither the Court in *Benson* or *Flook*, nor the CAFC in the FWA test clearly defined what precisely was covered by the term mathematical algorithm. See, e.g., *In re Warmerdam*, 33 F.3d 1354, 1359 (Fed. Cir. 1994); *Schrader*, 22 F.3d at 293 n.5. The CAFC equates mathematical algorithms with abstract ideas. See *State Street Bank*, 149 F.3d at 1373; see discussion *infra* notes 223-229, 302-311 and accompanying text.

non-trivial pre- or post-solution external activity, were certain to attract undesirable attention under the test.⁹¹ Any attempts to extend claims toward purer software-only inventions, therefore, held great possibility for misadventure before the PTO and even the CAFC.

Software development companies also sought to resolve the enforcement difficulties inherent in the PTO's and CAFC's treatment of most software inventions as claims to the underlying processes. Process patents are directly infringed only by use of the process. This meant that although a competitor might be distributing the enabling means (the software for implementing the process), only the actual end-user of the software who actually was implementing the process could be attacked as a direct infringer. Even if a patentee could overcome the commercial difficulty that the end-user was the patentee's own customer or potential customer, the problems of making multiple demands and bringing multiple suits against widely dispersed and hard to identify defendants remained. The only alternative was a more problematic contributory infringement claim against the competing software distributor.⁹² A more efficient and less problematic enforcement alternative was clearly desirable.⁹³

Inclusion of associated hardware platforms (a computer) or computer readable media carriers (diskettes, ROMs or the like) provided a vehicle for resolving both these difficulties. The theory was that by adding the computer or carrier, claims covering a software implemented process could be transformed into a "machine" (consisting of the software loaded in the computer) or an "article of manufacture" (the software loaded on the computer readable carrier). Because machines and articles of manufacture are expressly listed in § 101 as patentable subject matter, this reclassification would eliminate the application of the FWA test and its attendant difficulties.

Additionally, patents in these formats provided a means of direct enforcement against competitors. Because the claims expressly cover the software and diskette combination or the software pre-loaded into a computer, any competitor making or selling the software "bundled" in these

91. The battle over what constituted sufficient pre- or post-solution activity was a result of a need to satisfy the *Flook* holding that trivial post-solution activity was inadequate to escape the mathematical algorithm preemption test. See *Flook*, 437 U.S. at 589-90; *In re Christensen*, 478 F.2d 1392, 1394 (C.C.P.A. 1973); *In re Sarkar*, 588 F.2d 1330, 1335 (C.C.P.A. 1978); *Abele*, 684 F.2d at 907; *In re Meyer*, 688 F.2d 789, 796 (C.C.P.A. 1982); *Grams*, 888 F.2d at 839-40; *In re Schrader*, 22 F.3d 290, 293-94 (Fed. Cir. 1994).

92. Contributory infringement, at least arguably, requires some knowledge on the part of the defendant that the item being sold will be used for infringing purposes. See *Thomas*, *supra* at note 14.

93. IBM, the applicant in *In re Beauregard*, admitted as much in its court papers. See Richard H. Stern, *An Attempt to Rationalize Floppy Disk Claims*, *Oracle Symposium*, 17 J. Marshall J. Computer & Info. L. 183 (1998).

forms would directly infringe. The article of manufacture format will generally be the preferred alternative. In many situations the competitor will distribute the software separately from the overall computer system. In these cases the only infringement of a machine format claim would be by the end-user customer when she loads the software into the computer. The article of manufacture format, however, permits direct attack against a competitor's media based manufacturing and distribution activity.⁹⁴ More importantly, the article of manufacture format is the minimal claim format. In most cases the patent holder will be able to successfully argue that software on a disk or memory device incorporated in the computer infringes the article of manufacture claims, even though it is being sold as part of a complete system.

The "machine" version of these hardware-software combination formats was tested before the CAFC in *In re Alappat*.⁹⁵ Alappat's claims were written in "means for" terms.⁹⁶ Both the applicant and the PTO agreed the claims were broad enough to cover a general-purpose digital computer specially programmed to perform the specific task (line smoothing on a pixel matrixed oscilloscope screen).⁹⁷ Based on this reading of the claims, an eight member PTO review panel assembled for the reconsideration hearing (the PTO Board),⁹⁸ treated the application as *de facto* claiming of the underlying process.⁹⁹ It then applied the FWA/mathematical algorithm test to the process and found it involved non-statutory subject matter.¹⁰⁰ The CAFC, sitting *en banc*, rejected the PTO analysis.¹⁰¹ The court held that, properly interpreted in light of the specification's recitation of specific computing elements, the claims were limited to a "specific machine to produce a useful, concrete and tangible result"¹⁰² and as such, patentable subject matter.¹⁰³

The court's specific ruling concerning the claim coverage of a general

94. This depends on the competitor's use of some computer readable carrying medium. See discussion *infra* notes 275-277 and accompanying text.

95. *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994) (*en banc*).

96. See *supra* note 66 and accompanying text (describing "means for" claiming under 35 U.S.C. § 112, ¶ 6).

97. *Alappat*, 33 F.3d. at 1545.

98. In a very complex review, including having expanded the panel to eight members for the reconsideration hearing. *Id.* at 1531. These unusual procedural activities raised a question as to whether the CAFC had subject matter jurisdiction. The CAFC determined it did. *Id.*

99. *Id.* at 1539.

100. *Id.* at 1539-40.

101. The CAFC decision involved a number of separate dissenting and concurring opinions on both the jurisdictional issue and the substantive § 101 issue. The decision on the § 101 question garnered 5 votes for and 2 against, the remaining members of the court expressing no opinion due to their view that there was no jurisdiction over the case. See *supra* note 98 and accompanying text.

102. *Alappat*, 33 F.3d at 1544.

purpose computer running special purpose software¹⁰⁴ was, however, somewhat unclear. The court held that merely because claims read on a specially programmed general purpose digital computer does not automatically render them non-statutory: "a computer operating pursuant to software *may* represent patentable subject matter, provided of course, that the claimed subject matter meets all of the other requirements of [the Patent Act]."¹⁰⁵ Concerning when such claims actually are statutory, however, the court only provided cryptic guidance. The opinion does quote earlier CCPA and CAFC holdings that "[w]e 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."¹⁰⁶ Thus, in the court's view, at a minimum the introduction of the software creates a new machine. The opinion, however, left it ambiguous whether this newly created machine is itself statutory subject matter (*i.e.*, with the "other requirements" referring only to novelty, nonobviousness, definiteness and usefulness), or if something additional is required under § 101 and, if so, what that something more might be.¹⁰⁷

In 1998, the CAFC had a second opportunity to address the "machine" format in *State Street Bank & Trust v. Signature Financial Group*.¹⁰⁸ The case involved claims to a computer system for making allocations of assets, income, gains and expenses in a multi-member mutual fund partnership.¹⁰⁹ The United States District Court for the District Massachusetts had originally heard the case as a declaratory judgment action.¹¹⁰ That court held the patent invalid on patentable subject matter grounds. The court found that although the claims were written in "means for" apparatus format, they *de facto* covered the underlying process.¹¹¹ It went on to hold that process involved a mathematical algorithm which did not satisfy the FWA test.¹¹²

103. The hardware approach having the desired effect, with the court adopting an expressly statutory machine interpretation under § 101 and dismissing the "process"/FWA approach taken by the PTO Board. *Id.* at 1542-44.

104. *Id.* at 1544-45.

105. *Id.* at 1545. (*italics in original*)

106. *Id.*

107. *See Stern, supra* note 93 (indicating that in addition some kind of "connection" of the specially programmed computer to a specific application may be required).

108. *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368 (Fed. Cir. 1998).

109. *Id.* at 1369.

110. *State Street Bank & Trust v. Signature Fin. Group*, 927 F. Supp. 502 (D. Mass. 1996) (granting summary judgment for bank), *rev'd*, 149 F.3d 1368 (Fed. Cir. 1998).

111. *See id.* at 1370.

112. *Id.* at 1373-74. The court also held that the patent failed to claim statutory subject matter under the "method of doing business" exception. *Id.* at 1375. *See discussion infra*

The CAFC reversed. Applying the same approach used in *Alappat*, the court read the supporting structure contained in the specification into the respective means stated in the claim and held that the patent “properly construed, claims a machine” and “[a] ‘machine’ is proper statutory subject matter under § 101.”¹¹³ Addressing the lower court’s mathematical algorithm analysis, the CAFC found that the mathematical algorithm doctrine actually raised abstract idea concerns and should be reviewed accordingly.¹¹⁴ As a result, the critical issue was whether the algorithm/abstract idea had been “applied in a ‘useful’ way,” that is to produce “a useful, concrete and tangible result.”¹¹⁵ The court went on to hold that as the transformations performed by the claimed computer system did produce such a result,¹¹⁶ the patent involved patentable subject matter.¹¹⁷ The court noted that in this analytic framework “the Freeman-Walter-Abele test has little, if any, applicability to determining the presence of statutory subject matter”¹¹⁸ and, in particular, should not have been applied in this case.¹¹⁹

The *State Street Bank* decision has two important ramifications for the hardware claims approach. First, the opinion appears to resolve the *Alappat* issue concerning the status of a specially programmed computer under § 101¹²⁰ by finding that such systems are patentable subject matter only when they produce a “useful, concrete and tangible result.”¹²¹ Second, although the “output” analysis imposes some problematic and unjustified restrictions on use of the format,¹²² the decision resoundingly

notes 208-211 and accompanying text (discussing the method of doing business exception and the CAFC’s reversal eliminating the doctrine.)

113. *Id.* at 1375. Compare *In re Alappat*, 33 F.3d 1526, 1555 (Fed. Cir. 1994).

114. *State Street Bank*, 149 F.3d at 1373.

115. *Id.*

116. *Id.* at 1373 (“a final share price momentarily fixed for recording and reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades”).

117. *Id.* at 1373-74.

118. *Id.* at 1374.

119. *Id.*

120. See discussion *supra* notes 104-107 and accompanying text.

121. See *State Street Bank*, 149 F.3d at 1374. (“For purposes of our analysis [the claim] is directed to a machine programmed with [software] and admittedly produces a ‘useful, concrete, and tangible result. . . . This renders it statutory subject matter’”); *id.* at 1373-74 (“The plain and unambiguous meaning of § 101 is that any invention falling within one of the four stated categories of statutory matter may be patented, provided it meets the other requirements for patentability set forth in Title 35, e.g., those found in § 102, § 103, and § 112”).

122. The applicant and the examiner are left to come to terms on what precisely is meant by the “useful, concrete and tangible” result requirement. See discussion *infra* note 317 and accompanying text (discussing the confusion which can arise just from the tangibility requirement). In all events, the *State Street Bank* result is extremely difficult to square with *Alappat*’s citation to numerous cases reflecting the much more straight-for-

signals the success of the machine claim format in avoiding the uncertainties of the mathematical algorithm and FWA test.¹²³

C. "ARTICLE OF MANUFACTURE" CLAIMS IN THE CAFC AND THE PTO

The CAFC's willing abandonment of the FWA/mathematical algorithm analysis in the machine claim format context set the tone and direction for its approach to the second hardware-software combination claim format: software as article of manufacture. The article of manufacture approach defines the invention as a "software program product" consisting of a computer readable medium, called out as a diskette, ROM, DRAM or comparable piece of hardware in the detailed description, which "contains" a software program (set of instructions) designed to perform a particular task.¹²⁴ As noted earlier, if this format holds up, it is preferable to providers of software products. It offers the same escape from the mathematical algorithm/FWA difficulties and provides better enforcement options than the machine format.¹²⁵ The CAFC has had three opportunities to deal with the article of manufacture approach, in various degrees of directness, in *In re Warmerdam*,¹²⁶ *In re Lowry*¹²⁷ and *In re Beauregard*.¹²⁸

In re Warmerdam involved "a further refinement of prior art bubble systems" used as part of a collision avoidance mechanism.¹²⁹ The basic

ward *Prater dicta* approach (see *supra* note 73 and accompanying text) the corresponding language from *In re Bernhart*;

"[W]e say that if a machine is programmed in a certain new and unobvious way, it is physically different from the machine without the program; its memory elements are differently arranged. . . . If a new machine has not been invented, certainly a "new and useful improvement" of the unprogrammed machine has been, and Congress has said in 35 U.S.C. § 101 that such improvements are statutory subject matter for a patent."

In re Bernhart, 417 F.2d 1395, 1400 (C.C.P.A. 1969), the evolution in the article of manufacture cases, particularly *In re Lowry*, 33 F.3d 1579 (Fed. Cir. 1994) (see *infra* notes 152-155 and accompanying text) and the CAFC's en banc remand of *Trovato* citing *Alappat* and *Lowry* apparently indicating that the specially programmed computer's electronic structure should be enough to resolve that case. *In re Trovato*, 60 F.3d 807 (Fed. Cir. 1995)(en banc). The reconciliation lies in distinguishing, as the *State Street Bank* decision does not, between claims to software as apparatus and software as process. See *infra* note 229 and accompanying text.

123. See discussion *supra* notes 90-91 and accompanying text.

124. There are many permutations on the claim language. For example, the claims may use "embodied therein," "bearing," or "storing" to described the relationship between the medium and the software. See *infra* Appendix (setting forth sample claims). Despite these variations, the substance is as described in the text.

125. See discussion *supra* note 94 and accompanying text.

126. *In re Warmerdam*, 33 F.3d 1354 (Fed. Cir. 1994).

127. *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994).

128. *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995).

129. See *Warmerdam*, 33 F.3d at 1355.

concept involved using “nested” hierarchical bubbles enclosing the object to be avoided, with each succeeding internal level of bubbles more closely defining the exact contours of the object. Warmerdam’s application included six claims with four of the claims being process claims. None of the process claims were in software article of manufacture format. The process claims set out methods for constructing the bubble hierarchy, provided a claim to a bubble system “data structure” generated by any of the methods set out in the four process claims and disclosed a claim to “a machine having a memory which contains data representing a bubble hierarchy generated by any of the four process claims.”¹³⁰

On appeal the Board had sustained the examiner’s rejections, finding as follows: the process claims “recited no more than a mathematical algorithm in the abstract and thus failed to [claim statutory subject matter],”¹³¹ the claimed data structure “is not within one of the categories of patentable subject matter,”¹³² and the machine claim was indefinite under § 112, paragraph 2 (which requires a claim to clearly state and specify the invention) because “it left ‘unclear and unexplained how a memory is made or produced by the steps [in the method claims].”¹³³

Addressing the method claims first, the CAFC panel noted that applying the mathematical algorithm approach raised a variety of problematic definitional issues, not the least of which is determining what precisely constitutes a “mathematical algorithm.”¹³⁴ The panel found it unnecessary to deal with the problem, instead holding that the methods were unpatentable abstract ideas: “[The methods] describe nothing more than the manipulation of basic mathematical constructs, the paradigmatic ‘abstract idea.’ As the Supreme Court has made clear ‘[a]n idea of itself is not patentable,’ *Rubber-Tip Pencil Co. v. Howard*, 87 U.S. (20 Wall.) 498, 507 (1874); taking several abstract ideas and manipulating them together adds nothing to the basic equation.”¹³⁵

The court also upheld the rejection of the “data structure” claim, holding that “[s]ince the ‘data structure’ . . . is nothing more than another way of describing the manipulation of ideas contained in the

130. *Id.* at 1357-58 (the quoted language is from Claim 5). It is important to note that except for Claim 5, the machine claim, and Claim 6, the data structure claim, Warmerdam’s claims are stated expressly in general “method” terms, not as claims to software implementations of those methods. Claim 6 also claimed the data structure generally without limitation to a computer environment.

131. *Id.* at 1558.

132. *Id.*

133. *Id.*

134. *Id.* at 1359. This was a key reason patent applicants claiming software inventions sought to avoid that test. See discussion *supra* note 91 and accompanying text.

135. *Id.* at 1360. See discussion *infra* notes 223-225 and accompanying text (discussing the abstract idea exclusion under § 101).

[method] claims, it suffers from the same fatal defect they do.”¹³⁶ The court specifically distinguished Warmerdam’s data structure from the claims in *In re Bradley*¹³⁷ on the grounds that in this case the claims did not involve an “arrangement of hardware.”¹³⁸

Regarding the “machine” claim the court started its analysis by noting in dicta that the claim “is for a machine, and is clearly patentable subject matter.”¹³⁹ The court then reversed the Board’s § 112 indefiniteness rejection. The court found “[t]he legal standard for definiteness is whether a claim reasonably apprises those of skill in the art of its scope.”¹⁴⁰ Applying this test to the claim, the court held

“[t]he ideas in [the method claims] are well known mathematical constructs, and lend themselves to manipulation through known computer technology. There has been no showing that one skilled in the art would have any particular difficulty in determining whether a machine having a memory containing data representing a bubble hierarchy is or is not within the scope of [the claim].”¹⁴¹

136. *Id.* at 1362.

137. *In re Bradley*, 600 F.2d 807 (C.C.P.A. 1979), *aff’d by equally divided court, sub nom.* *Diamond v. Bradley*, 450 U.S. 381 (1981) (Burger, C.J., not participating). It seems that claims which the court views as involving only “logical” structure (Warmerdam’s data structure) will have considerably more difficulty passing muster as patentable subject matter than claims including “physical” (hardware) structure, despite the “physical or logical” phraseology in the definition of data structure used by the court. *Id.* The point is picked up in *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994) which appears to find the physical changes resulting from the data structure model in that case are sufficient to make it statutory. See discussion *infra* notes 149-155 and accompanying text.

138. *Warmerdam*, 33 F.3d at 1362.

139. *Id.* at 1360. The *Warmerdam* panel appeared relatively untroubled by the possible position that *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994), requires something beyond a specially programmed general purpose computer to satisfy § 101. See discussion *supra* notes 104-107 and accompanying text. The panel’s discussion of § 112 does end by noting “[w]hether such a programmed machine is new, useful, unobvious or otherwise patentable is not at issue in this appeal.” *Id.* at 1760. Although the “otherwise patentable” language seems a bit vague, when read against earlier “machine” dicta it appears to contemplate requirements beyond those necessary to satisfy § 101. In all events, the position was clarified in *State Street Bank*. See discussion *supra* notes 108-123 and accompanying text.

140. *Id.* at 1361.

141. *Id.* Another interesting issue and complexity introduced by the article of manufacture claim format is the “product by process” question. In *Warmerdam* the question is raised, although the CAFC justifiably questions whether the issue arises on the particular facts. *Id.* at 1361 n.6. Had Warmerdam claimed the data structure on computer readable media, the issue would need to be confronted, as the claimed data structure contains the product of the claimed bubble system creation, thereby leading directly into the considerable controversy at the CAFC regarding proper review of such claims under §§ 102 and 103. The alternative approach proposed by this article avoids the issue entirely by eliminating the confusion introduced by focusing on the presence or absence of computer readable media. See *infra* note 242 and accompanying text.

Because the applicant did not use a software as article of manufacture claim structure, *Warmerdam* does not provide much direct indication of the CAFC's view of that particular format. It does, however, provide interesting insights into this panel's thought processes on two relevant issues. First, the different view of the machine claim¹⁴² and the data structure claim¹⁴³ reinforces the use of hardware "environments" in software claims as an effective way to clear the § 101 requirement. Second, the definiteness discussion notes that software claims should be examined under § 112 based on the same "skill in the art" standard applied to other categories of invention.

The second case, *In re Lowry*, also involved a "data structure." The invention was based on an "attributive data objects" ("ADO") model which permits information about data to be extracted and described in terms of both its characteristics and its relationship with other data. Unlike *Warmerdam*'s general data structure claim, however, *Lowry*'s key claims were couched in the following terms: "a memory for storing data for access by an application program being executed on a data processing system" with the data arranged according to the ADO model.¹⁴⁴

The examiner rejected the claims as non-statutory subject matter and for obviousness. On appeal the PTO Board reversed the non-statutory subject matter claim, finding instead that the claims "directed to a memory containing stored information, as a whole, recited an article of manufacture" and was statutory subject matter.¹⁴⁵ On the obviousness rejection, the PTO Board analogized the data contained in the memory device to printed matter and found that when "printed matter is not functionally related to the substrate, the printed matter will not distinguish the invention from the prior art in terms of patentability."¹⁴⁶ The PTO Board, therefore, refused to consider the content of the memory, in particular the special organization and structure of the data according to the ADO model, and upheld the obviousness rejection over prior art which essentially disclosed "a CPU using a memory and containing stored data in a data structure."¹⁴⁷

On appeal, the CAFC panel found the PTO Board's printed matter analogy inapplicable on two grounds. First, "[t]he printed matter cases have no factual relevance where 'the invention as defined by the claims

142. See discussion *supra* note 139 and accompanying text.

143. See discussion *supra* notes 136-138 and accompanying text.

144. See *Lowry*, 32 F.3d at 1581.

145. *Id.* at 1582. This is consistent with the dicta concerning Claim 5 in *Warmerdam*, 33 F.3d at 1360; see discussion *supra* notes 142-143 and accompanying text.

146. *Lowry*, 32 F.3d at 1582 (citing *In re Gulack*, 703 F.2d 1381 (Fed. Cir. 1983)).

147. *Id.* Clearly on this basis virtually all software as article of manufacture claims will be obvious over the prior art of placing software programs on a computer readable medium.

requires that the information be processed not by the mind but by a machine, the computer'.¹⁴⁸ Second, the court found that the claims were not to the specific information content of the data structure, but to "the specific electronic structural elements which impart a physical organization on the information stored in memory."¹⁴⁹ The court's holding on this point is worth citing in full:

More than mere abstractions, the data structures are specific electrical or magnetic structural elements in memory. According to Lowry, the data structures provide tangible benefits: data stored in accordance with the claimed data structures are more easily accessed, stored and erased. . . . In short, Lowry's data structures are physical entities that provide increased efficiency in computer operation. They are not printed matter. The Board is not at liberty to ignore such limitations.¹⁵⁰

Having eliminated the "printed matter" underpinnings for the PTO Board's rejection, the court easily found that the specific organization of the data according to the claimed data structure was distinguishable over the cited prior art and reversed the PTO Board's obviousness finding.¹⁵¹

Although *Lowry* does not specifically involve a software program claimed on computer readable media as an article of manufacture, several key points are relevant to the article of manufacture approach. First, the CAFC panel does not object to, in dicta, footnote or otherwise, the PTO Board's characterization of the claimed invention as an article of manufacture rather than a process. The case did not arise in a vacuum of first impression. Certainly the panel was aware of the effect under § 101 of the PTO Board's classification. Even though the issue was arguably not technically before the court, if the panel had been concerned about the issue it had ample opportunity to so indicate. Quite to the contrary, the panel's analysis, resting on the physical changes in the medium caused by the data structure, is entirely consistent with classification of the claimed invention as an article of manufacture. Thus,

148. *Id.* at 1583 (citing to *In re Bernhart*, 417 F.2d 1395, 1399 (C.C.P.A. 1969) in which the CCPA had told the PTO exactly the same thing 25 years earlier). (italics in original)

149. *Id.*

150. *Id.* at 1583-84. The CAFC also noted that Lowry was not claiming the data model in the abstract, but the "physical organization" it imposed on the data in the memory. *Id.* at 1034. Under the analytical framework proposed in this article even a claim to the model itself, when confined to a software instruction computer system implementation, should be statutory as well. See discussion *infra* Part III.C.

151. *Lowry*, 32 F.3d at 1584. This is a good example of how the current software as article of manufacture analysis distorts the § 102 and § 103 prior art identification issue and examination. Defining the applicable prior art is not addressed by the panel in *Lowry*. Should the examiner look for prior art concerning an advance in data organization generally, application of to the model to computer technology, the model's automation, or all three?

Lowry reinforces the general efficacy of including computer hardware in the claims as a method for avoiding FWA/mathematical algorithm § 101 review.

Second, the *Lowry* data structure receives much different treatment than the ill-fated data structure claim in *Warmerdam*. Citing *In re Bernhart*,¹⁵² the *Lowry* panel expressly indicated its willingness to find that data structures are physical entities based on the electronic or magnetic arrangement in the computer memory even though this physical structure is "invisible to the eye."¹⁵³ This makes it appear that the difficulty with *Warmerdam*'s data structure claim may have been *Warmerdam*'s failure to put it on an electronic medium. So modified, at least under the "physical structure" logic of *Lowry* and the dicta concerning *Warmerdam*'s machine claim, *Warmerdam*'s data structure claim could be transformed from the mere manipulation of "abstract ideas" into an apparatus consisting of the specific physical electronic structure of the memory reflecting the incorporated bubble structure.¹⁵⁴

Although it is not the holding in either *Warmerdam* or *Lowry*, these cases provide reasonable support for the following article of manufacture claim argument: if the claimed implementation causes changes to the

152. *In re Bernhart*, 417 F.2d 1395 (C.C.P.A. 1969).

153. *Lowry*, 32 F.3d at 1583. This could be read as providing the necessary physical structure to avoid the *Warmerdam* data structure holding. Viewed in this light *Lowry* seems to support the view that when specific software is loaded into a general purpose digital computer the resulting special computer, standing alone, is statutory subject matter. Certainly if changing the physical structure of a diskette or ROM by loading it with data organized in a fashion consistent with *Lowry*'s data model makes that invention a statutory article of manufacture the changes caused by loading the software into a specially programmed computer must create a statutory machine. One can also argue, however, that the result is consistent with *State Street Bank*; the resulting data structure being a "useful, concrete and tangible result." See discussion *supra* notes 104-123 and accompanying text. The issue can be resolved by untangling the apparatus-process confusion in *State Street Bank*. See *infra* note 229 and accompanying text.

154. There is, however, a critical difference between the cases which is not resolved by this proposed analysis. Unlike *Lowry*'s claim which involves software implementation of a general computer system storage methodology, *Warmerdam*'s data structure on computer readable media would merely consist of the bubble data resulting from application of the bubble hierarchy methodology. Approving *Warmerdam*'s article of manufacture data structure claim might, therefore, permit the particular data resulting from the bubble methodology (clearly not patentable subject matter) to satisfy § 101. See *infra* note 220 and accompanying text. The *Guidelines*, *supra* note 4, would likely reject this form of *Warmerdam* claim as non-functional descriptive material (see discussion *infra* notes 170-178 and accompanying text). Under the *Guidelines* what *Warmerdam* must claim instead is the software for creating the data structure, not the resulting data, on computer readable media. *Id.* A counter argument could trigger the complexity of product by process analysis (it is not the specific data but the data resulting from the bubble creation process which is claimed). The complexity of this entire exercise leaves considerable room for error. A different means for reconciling *Warmerdam* and *Lowry* is required. See *infra* note 155 and accompanying text.

electronic structure of the included computer readable medium, then the claim contains physical structure. If the claim contains physical structure, then it involves patentable subject matter, either a machine or an article of manufacture, under § 101.¹⁵⁵ There is, therefore, no need to apply the mathematical algorithm based FWA test.

The final case, *In re Beauregard*, presented the software as article of manufacture format directly by claiming an article "composed of a computer usable medium in which a program code is embodied."¹⁵⁶ The software program implemented a method for filling-in complex shapes on a computer screen.¹⁵⁷ The case posed the "next step" question, if under *Lowry* a data structure on computer readable media is § 101 patentable subject matter as an article of manufacture, why not a software program on computer readable media?

The PTO Board of Appeals had rejected by a 4-3 vote all the claims using a variant of the printed matter analysis used in *Lowry*.¹⁵⁸ The CAFC never heard the case. Faced with the decision in *Lowry*, the PTO conceded "that computer software programs embodied in a tangible medium such as floppy diskettes, are patentable subject matter under [§ 101]."¹⁵⁹ The CAFC, noting the parties agreed this concession eliminated any present case or controversy, vacated the PTO Board's rejection and "remanded for further proceedings in accordance with the Commissioner's concessions."¹⁶⁰

155. Although a reasonable reading, this formulation overstates the actual requirement in *Lowry*, focusing too much on the media and missing the requirement that the information be processed by a machine (the computer). *Lowry*, 32 F.3d at 1583. The two cases can be better reconciled on this latter basis; recognizing that *Lowry* limited his claims to computer system implementation whereas *Warmerdam* did not. This reconciliation reflects the implementation-language dichotomy test proposed in this article. See discussion *infra* notes 254-258 and accompanying text.

156. PTO Board of Patent Appeals and Interferences, Appeal No. 93-0378, Heard June 17, 1993, at 1-2 [hereinafter *PTO Board*]. The actual claim read "An article of manufacture comprising: a computer usable medium having a computer readable program code means embodied therein for causing [the filling activity]." See *infra* Appendix, Part II. *Beauregard* had already received a computer system and process patents claims to the process implemented by the computer program. *Id.* at 2.

157. *Id.* at 2.

158. *Id.* at 5-7. The two cases were on parallel tracks in the PTO with *Lowry* making its way to the CAFC first.

159. *In re Beauregard*, 53 F.3d 1584 (Fed. Cir. 1995). The PTO Board had expressly rejected applicant's argument that *In re Bernhart*, 417 F.2d 1395 (C.C.P.A. 1969) controlled, finding that in *Bernhart* "the programmed instructions were presently used as part of the method and apparatus claimed. No such present use is recited by these claims." *PTO Board*, *supra* note 156, at 11. The *Lowry* decision, which relied on *Bernhart*, indicated it was unlikely the PTO Board's position would be sustained by the CAFC. See discussion *supra* notes 148-150 and accompanying text.

160. *Beauregard*, 53 F.3d at 1584.

D. THE PTO GUIDELINES

Faced with the decisions in *Alappat*, *Warmerdam* and *Lowry*, the PTO had capitulated. In February 1996 the PTO issued its final Examination Guidelines for Computer-Related Inventions¹⁶¹ in which it stated it was adopting the CAFC's approach to software inventions as the guiding principles for the examination.¹⁶²

The Guidelines do not deal specifically with software as article of manufacture claims. They set out the following generally applicable six-step approach¹⁶³ to examination of all software-related claims:

1. Determine what the applicant has invented and is seeking to patent.
2. Conduct a thorough search of the prior art.
3. Determine whether the invention claims patentable subject matter (§ 101 review).
4. Evaluate the application for compliance with § 112 requirements.
5. Determine whether the claimed invention is novel (§ 102) and non-obvious (§ 103).
6. Clearly communicate the findings, conclusions and their bases to the applicant.¹⁶⁴

To understand the PTO's position concerning software as article of manufacture claims requires review of steps 1, 3 and 4 of the software examination process.

Step 1 expressly states that the examiner "will no longer begin examination" by determining if a claim recites a "mathematical algorithm."¹⁶⁵ Clearly, that analytical structure has been demoted under

161. *Guidelines, supra* note 4. The PTO had previously issued draft Guidelines for comment. 60 Fed. Reg. 28,778 (1995). That draft was withdrawn and the substantially revised final *Guidelines* issued without further request for comment.

162. *Guidelines, supra* note 4, at 7479. The PTO pointed out that as the *Guidelines* were merely tracking existing law they were not substantive rule-making. They, therefore, cannot be cited as legal authority in their own right. *Id.* This did not stop citation to the *Guidelines* for support either by the CAFC in *In re Trovato*, 60 F.3d 807 (Fed. Cir. 1995) (en banc) even though at the time they were then only in proposed form (much to the dismay of the dissent, *Id.* at 808 (Nies, J., dissenting) or the first District Court addressing the issue, ruling in *State Street Bank & Trust v. Signature Fin. Group, Inc.*, 927 F. Supp. 502, 512 (D. Mass. 1996), *rev'd*, 149 F.3d 1368 (Fed. Cir. 1998).

163. An actual flowchart is set out as an Appendix to the *Guidelines*. *Guidelines, supra* note 4, at 7491-92.

164. In the past, examination frequently would reach the § 101 rejection and stop. This had two undesirable consequences. First, from a practice standpoint it could mean a number of future rounds with the examiner once the § 101 hurdle was cleared (sometimes involving an appeal). Second, in many instances, if PTO lost the fight on the § 101 issue the examiner would clear the invention without significant § 102 or § 103 examination. The *Guidelines* now instruct the examiner to "state all reasons and bases for rejecting claims in the first Office action." *Id.* at 7479 (as called for by the CCPA in 1979 in *In re Phillips*, 608 F.2d 879, 883 n.6 975 (C.C.P.A. 1979). This is a notable improvement.

165. *Guidelines, supra* note 4, at 7479. The *Guidelines* also note "The Freeman-Walter-Abele test may additionally be relied upon in analyzing claims directed solely to a process

the new regime. Instead, the examiner is to "review the complete specification, including the detailed description of the invention . . . the claims and any specific utilities that have been asserted for the invention."¹⁶⁶ This Step's objective is to understand the claimed invention as a whole, including applicable limitations on the claims, so the subsequent examination will be properly focused.¹⁶⁷ The step clearly rejects any "point of novelty" parsing or breaking down of the claim, focusing the examiner on the invention's full scope and limitations.¹⁶⁸

The § 101 inquiry as laid out in Step 3 seeks to classify the invention either into one of the four listed § 101 statutory categories (processes, machines, manufactures or compositions of matter) or one of three non-statutory categories (natural phenomena, abstract ideas or laws of nature).¹⁶⁹ The process for accomplishing this classification, reflecting the PTO's *Beauregard* concession on program code, contains a distinctly positive disposition toward software when claimed in article of manufacture format, generally classifying them as statutory subject matter.

To ensure that the favorable treatment afforded computer programs does not spill over to permit patentability of computer data, which quite properly remains extremely suspect under § 101, the PTO needed to distinguish between the two.¹⁷⁰ The Guidelines address the problem by creating a classification scheme based on a general definitional rubric of

for solving a mathematical algorithm." *Id.* That test, therefore, remains, but with narrower application.

166. *Id.* This latter reference is consistent with the Step 1 guidance that the examiner should seek to understand the "usefulness" of the invention in terms of its "real world" value. *Id.* ("The purpose of this requirement is to limit patent protection to inventions that possess a certain level of "real world" value, as opposed to subject matter that represents nothing more than an idea or concept") This is to identify inventions which are really claiming "no more than an idea or concept." *Id.* Additionally, the utility must be "within the technological arts," which the *Guidelines* concede will always be the case with computer-related inventions provided they have a practical ("real world") application. *Id.* This approach is on the right track, but falls short of the mark. *See* discussion *supra* Part III.C.

167. *Guidelines*, *supra* note 4, at 7480.

168. *See* discussion *supra* notes 58-63 and accompanying text.

169. *Guidelines*, *supra* note 4, at 7481. The three non-statutory categories are considered to be exclusive. *Id.* This is primarily a reaction to *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994) which indicated the Supreme Court had not intended in *Gottschalk v. Benson*, 409 U.S. 63 (1972); *Parker v. Flook*, 437 U.S. 584 (1978); or *Diamond v. Diehr*, 450 U.S. 175 (1981) to create a fourth category of non-statutory subject matter, *i.e.*, mathematical algorithms. *See Alappat*, 33 F.3d at 1543. This clearly overstates the case as all "non-useful arts" inventions, for example, music, plays and novels, as such are also not patentable subject matter. Recapturing this omission, the *Guidelines*, in a non-sequiter, creates a class of non-functional descriptive material much broader than the three excluded classes to cover such creations. *Guidelines*, *supra* note 4, at 7482.

170. This line drawing concern was raised by the dissent in *Alappat*, 33 F.3d at 1554 and 1556 and the PTO Board majority in *Beauregard*. *PTO Board*, *supra* note 156, at 16. The concern is that an inability to draw this line would result in music and other data

“descriptive material.” Descriptive material is divided into “functional” and “non-functional” categories.¹⁷¹ Functional descriptive material consists of “data structures and computer programs which impart functionality when encoded on a computer-readable medium.”¹⁷² Non-functional descriptive material essentially consists of “everything else,” including data.¹⁷³

Each type of descriptive material is assessed under the requirements of § 101. Because non-functional descriptive material (including computer readable data) is never able to impart functionality, it is always treated as non-statutory subject matter.¹⁷⁴ Consistent with the concession in *Beauregard*, the Guidelines classify functional descriptive material, software and data structures which can impart functionality, as statutory or non-statutory depending on whether the claims include computer readable media:

[C]omputer programs claimed as computer listings per se . . . are not physical ‘things,’ . . . as they are not ‘acts’ being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed aspects of the invention which permit the computer program’s functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program defines structural and functional interrelationships between the computer program and the medium which permit the computer program’s functionality to be realized, and is thus statutory.¹⁷⁵

The key is permitting implementation of the software functionality. Because claims to a computer program standing alone (a listing) do not permit the functionality to be realized, such claims are non-patentable subject matter. However, claiming the same software on computer readable media, thereby permitting its incorporated functionality to be read

becoming patentable virtually *per se* by claiming it on media of some form. See *infra* note 242 and accompanying text.

171. *Guidelines*, *supra* note 4, at 7481-82.

172. *Id.* at 7481.

173. The *Guidelines* set out a non-inclusive list of examples. *Id.* at 7481-82. Most important are music and literary works addressing the specific examples raised in the *Alapat* dissent and by the PTO Board majority in *Beauregard*. See *supra* note 170 and accompanying text. So are computer program listings or data structures claimed as such. See *infra* notes 175-176 and accompanying text.

174. *Guidelines*, *supra* note 4, at 7481. The *Guidelines* also note that because such material will not be structurally or functionally interrelated to the media so as to permit the functionality to be realized, merely placing it on computer readable media will not make it statutory subject matter. *Id.* Although this sounds suspiciously like a printed matter rejection the *Guidelines* justify it in this context as necessary to avoid “exalt[ing] form over substance.” *Id.*

175. *Id.* at 7481-82.

by the computer and implemented, makes the combination statutory.¹⁷⁶

Although the Guidelines expressly confirm that software presented in an article of manufacture form is generally statutory, the last step in the § 101 analysis revisits the "smuggling in" concerns and carves out process claims masquerading as apparatus claims. Only if the claims detail a specific machine or manufacture should they be viewed as true apparatus claims and expressly statutory.¹⁷⁷ If the claims are written to cover "any and every" machine or manufacture they should instead be examined as *de facto* claims to the underlying process steps¹⁷⁸ requiring further analysis to determine whether that process involves statutory subject matter.¹⁷⁹

Despite considerable effort at explaining the difference between claims to a "specific machine or manufacture" and claims to "any and every machine or manufacture," the distinction is not entirely clear.¹⁸⁰ The basic requirement for finding that the former is present appears to

176. Although headed in the right direction, the test in the *Guidelines* fails to draw the necessary line between the software being used as language and as technical implementation. See discussion *supra* Part III.C.

177. *Guidelines*, *supra* note 4, at 7483.

178. *Id.* at 7482. The *Guidelines* cite (as a "*cf*" which supports a proposition different from the main proposition but sufficiently analogous to lend support) to *In re Iwahashi*, 888 F.2d 1370, 1374-75 (Fed. Cir. 1989) in support of the proposition that merely including hardware in the claims does not indicate the presence of a specific machine or manufacture. *Guidelines*, *supra* note 4, at 7482 n.37. Interestingly in *Iwahashi*, the PTO took this position and lost based on the CAFC's finding that specific hardware was in fact claimed. *Iwahashi*, 888 F.2d at 1375. The general statement of the "any and every" proposition is found in *In re Freeman*, 573 F.2d 1237 (C.C.P.A. 1978), in which the CAFC held "we agree with the solicitor's contention that if allowance of a method claim is proscribed by *Benson*, it would be anomalous to grant a claim to apparatus encompassing any and every "means for" practicing that very method." *Freeman*, 573 F.2d at 1247. In *In re Maucorps*, 609 F.2d 481, 486 (C.C.P.A. 1979) the CCPA actually agreed with the PTO and rejected claims based on the "any and every means" test. Note, however, that claiming all computer implementations of a process is *not* the same as claiming any and every means for implementing a process. The former is just a claim to the computer implementations of a process; others are free to implement the process in any other fashion. Interpreting this, as the *Guidelines* do, as preemption of the entire process smacks of the discredited preemption of a mathematical algorithm test from *Benson*. See, e.g., *In re Waldbaum*, 59 F.2d 611, 616-17 (C.C.P.A. 1977) (relying on *Benson*). See discussion *infra* notes 269-271 and accompanying text.

179. Process claim examination under the *Guidelines* retains much of the flavor of pre-Guideline software review, a modified mathematical algorithm *Freeman-Walter-Abele* review, complete with safe harbors based on the CAFC cases applying the FWA test. *Guidelines*, *supra* note 4, at 7479, 7483-84. A final test permits processes to pass muster if they are "limited to a practical application of the abstract idea or mathematical algorithm in the technological arts." *Id.* at 7484. Ironically, the *Guidelines* fail to recognize that when limited to its computer system implementation role, the claimed software itself is always just such an application.

180. *Id.* at 7482-83.

be that the claims define the "physical structure of the machine or manufacture in terms of its hardware or hardware and 'specific software.'"¹⁸¹ Because an article of manufacture claim will normally include every form of computer readable media, the primary question becomes whether the claims provide enough specifics concerning the operation of the "carried" software.¹⁸² Based on the example provided in the Guidelines, claims which include sufficient detail (either in actual program code or logic circuits) fall on the "specific manufacture" side and those which leave implementation primarily to "those of ordinary skill in the art" on the "process" side. The contest between applicant and examiner will come down to determining the minimum amount of specific detail required to satisfy the "specific manufacture" requirement.¹⁸³

Finally, step 4 of the Guidelines takes a fairly traditional approach to the § 112 requirements and will generally be easy to satisfy. Of interest are two references to the discussion which emphasize that any software relied upon to meet the § 112 requirements should also conform to the § 101 article of manufacture requirements. First, in the case of "means for" style claims if "the applicant discloses only the functions to be performed and provides no express, implied or inherent disclosure of hardware or a combination of hardware and software that performs the functions,"¹⁸⁴ a rejection under § 112, paragraph 2¹⁸⁵ is appropriate for failure to particularly point out and distinctly claim the invention. Reflecting the § 101 analysis, the Guidelines indicate that if software is to serve as the disclosed means, the software must be in article of manufac-

181. *Id.* at 7483.

182. *Id.* ("To adequately define a specific computer memory, the claim must identify a general or specific computer memory and the specific software which provides the functionality stored in the memory.")

183. *Id.* This problem will frequently arise in apparatus claims framed in "means for" style. See *supra* note 65 and accompanying text. Absent adequate software specifics such claims will be analyzed as process claims under the *Guidelines*. This was the issue in *Trovato* and, in part, the basis for the rejections under the original decision. See *In re Trovato*, 42 F.3d 1376, 1382-83 (Fed. Cir. 1994), *vac and remanded*, 60 F.3d 807 (Fed. Cir. 1995) (en banc). The *en banc* remand of *Trovato* indicated the case should be considered in light of *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994), *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994) and the then draft PTO *Guidelines* (see *supra* note 142 and accompanying text). *Trovato*, 60 F.3d at 807. Although, as the dissent in the remand notes, it was not particularly clear in the majority opinion precisely how any of those authorities might change the result, the inference must be that if the specification provided some form of computer readable medium (or other hardware) and adequate specifics concerning the software, the claims should be cleared under § 101. See *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1372-73 (Fed. Cir. 1998) (applying this approach). Of course, even if it passes muster under § 101, a "means for" claims which leaves virtually all implementation detail to those of "ordinary skill in the art" will have serious non-obviousness problems under § 103.

184. *Guidelines, supra* note 4, at 7486.

185. 35 U.S.C. § 112, ¶ 2 (1994).

ture/computer readable media form.¹⁸⁶ In addition, to avoid such a rejection, there must be sufficient detail regarding the software to permit one of skill in the art to determine the scope of the invention claimed.¹⁸⁷

Second, for purposes of enablement,¹⁸⁸ the examiner is instructed to "ensure that along with the functional block diagram, the disclosure provides information that adequately describes each 'element' in hardware or hardware and its associated software and how such elements are interrelated."¹⁸⁹ The "hardware and its associated software" again clearly emphasizes the article of manufacture claim structure. Regarding the description of the incorporated software, the applicant can elect either to expressly provide more detail or fall back on the "person of ordinary skill in the art."¹⁹⁰

The practical experience to date under the Guidelines indicates the PTO meant what it said. Examiners appear to be taking a literal and substantially more liberal view of the patentability of software related inventions claimed in article of manufacture form.¹⁹¹ Not surprisingly, the effect has been to shift applicants to use of this more favorable and less problematic format for claiming software-related inventions.

186. The *Guidelines* state that acceptable means include "a computer memory encoded with executable instructions representing a computer program that can cause a computer to function in a particular fashion." *Guidelines*, *supra* note 4, at 7486.

187. *Guidelines*, *supra* note 4, at 7486-87. The *Guidelines* encourage the applicant to describe the software in terms of "the steps the computer will perform" rather than "simply reciting source or object code." *Id.* at 7486. Taken literally that could send the applicant directly into the jaws of a § 101 disguised process claim determination by the examiner. See *supra* note 183 and accompanying text. The intent must be that the applicant should describe the special, specific functionality imparted by the carried software program so the examiner does not need to extract it from the code itself.

188. 35 U.S.C. § 112, ¶1 (1994).

189. *Guidelines*, *supra* note 4, at 7487 (noting "hardware and associated software" includes software on computer readable media). See *supra* note 187 and accompanying text.

190. *Guidelines*, *supra* note 4, at 7487. This approach raises § 101 risks. See discussion *supra* notes 177-183 and accompanying text.

191. No hard data has yet been assembled, but the general sentiment among members of the bar is that getting a patent on software claimed in article of manufacture form is becoming a relatively routine matter with virtually no 101 rejections being experienced. Certainly the practitioner's literature is filled with relatively straight-forward "how to" article of manufacture claiming documentation rather than discussions concerning how to deal with examiner rejections under § 101. See, e.g., Brian R. Yoshida, *Claiming Electronic and Software Technologies*, 45 *Buff. L. Rev.* 457, 495 (1997); S. Becker, *Preparing U.S. Patent Applications under the New Guidelines*, 10 *COMPUTER LAWYER* 10, 13 (1996); Jonathan E. Retsky, *Computer Software Protection in 1996: A Practitioner's Nightmare*, 29 *J. MARSHALL L. REV.* 853, 863 (1996).

E. WHAT'S WRONG WITH CURRENT ARTICLE OF MANUFACTURE PRACTICE? WHAT WE ARE TRYING TO ACHIEVE WITH THE PATENT SYSTEM

The new hardware-focused regime under the Guidelines provides a number of administrative benefits compared to the prior mathematical algorithm approach, even in its *Freeman-Walter-Abele* incarnation. From the applicant's perspective, the framework provides significant improvements in terms of certainty and related increased likelihood of a patent grant.¹⁹² The traditional jousting with the examiner concerning the scope of coverage, particularly regarding the degree of precision required in detailing specific software elements will, of course, continue. However, by following the formula set out in the Guidelines, an experienced patent practitioner ultimately should have little difficulty in obtaining a patent with acceptable claims for a computer program using the computer readable media/article of manufacture format. From the examiner's perspective, the system's simplicity and quasi-objectiveness makes case examination more focused and, consequently, a more efficient, faster process. Finally, patents issued in the hardware formats, particularly the article of manufacture variant, permit direct enforcement against competing software distributors.¹⁹³ Both software patentee difficulties under the former FWA/process approach appear to have been resolved.

Although efficient administration and enforcement are important, the primary objectives of the patent laws are not achieved merely by ensuring ease and speed of prosecution, predictability of results or maximizing enforcement options. To take the true measure of current practice requires an exploration of its consistency with the policy reasons for having a patent law at all.

As discussed in Part A above, the United States patent laws rest on the premise, embodied in the Constitution, that providing an economic incentive to inventors in the form of a right to exclude others for a limited term will lead to more inventions which, in turn, will lead to improvement in the general social condition.¹⁹⁴ Accepting for purposes of this discussion that the economic approach actually operates to achieve the desired objective of increased invention,¹⁹⁵ then the primary social

192. See discussion *supra* note 91 and accompanying text.

193. See discussion *supra* note 92, 94 and accompanying text.

194. See *supra* note 36 and accompanying text.

195. There are doubts about the efficacy of the economic model. See *Merges et al., supra* note 70, at 19-20 (citing sources). *But see, National Planning Comm. First Report* (1943) (lauding the contributions of the patent laws to U.S. growth); LIPSCOMB, *supra* note 34, § 1.9. As with many aspects of intellectual property law more empirical research would be useful. At a minimum, as a realist proposition, the recent GATT-TRIPS negotiations and

engineering issue is to determine which inventive activities, if any, should be targeted by the patent incentive.

This determination, in turn, requires identification of those situations where granting patent protection ultimately provides social prosperity returns in excess of costs arising from the resulting restrictions on competition.¹⁹⁶ This was the very issue addressed by Jefferson and Madison when debating whether to include the Patent Clause in the Constitution in the first instance.¹⁹⁷ Thomas Jefferson eloquently captured the essence of the inquiry: “[D]rawing a line between things which are worth to the public the embarrassment of an exclusive patent and those which are not.”¹⁹⁸ The present inquiry into the patentability of software as article of manufacture, therefore, requires a determination of whether this claim format falls on the “worth it” side when tested under Jefferson’s line-drawing exercise.

The Framers’ decision concerning the incentive-embarrassment tension provides a good starting point. Their resolution is clearly reflected in the Patent Clause which expressly limits congressional use of the patent power “to promote the progress of . . . [the] useful arts.”¹⁹⁹ This language establishes the basic policy (and legal) outer boundary condition concerning the software as article of manufacture question.²⁰⁰ If under

resulting treaty indicate the assumption continues to have significant adherents and that the view is gaining rather than losing strength.

In addition, one could argue as a philosophical matter, we simply have it all wrong and that other normative underpinnings for the patent laws, such as natural rights or personhood approaches, are more appropriate. Such a fundamental change in approach would alter the discussion of appropriate application of the patent incentive. This inquiry, of course, challenges the very premises of patent law and go far deeper than the question of the patentability of software.

196. See, e.g., *Graham v. John Deere Co.* 383 U. S. 1, 8-9 (1966); STOBBS, *supra* note 18, § 1.7; WILLIAM C. ROBINSON, *THE LAW OF PATENTS* § 3.3 (1890).

197. See *supra* note 34 and accompanying text.

198. IV Writings of Thomas Jefferson (Washington ed.) (1814). See *Graham*, 383 U.S. at 7-10 (discussing the Jeffersonian view).

199. U.S. CONST., art. I, § 8, cl 8.

200. This author is not aware of any recent calls to expand the reach of the patent laws beyond the useful arts. This does not mean that good policy arguments cannot be made that other forms of human activity also merit patent-like incentives. It merely means that if society believes additional areas of human endeavor deserve such incentives then the useful arts limitation contained in the Constitutional provision needs to be revisited, at least if it is assumed that the specific grant in the Patent Clause precludes additional action by Congress under its other powers such as the Commerce Clause. If the economic basis for the patent incentive is the starting premise, determining whether to extend the patent law incentive involves a calculation of the related costs and benefits. See, e.g., discussion *infra* note 221 and accompanying text (discussing the costs of extending patent protection to laws of nature).

Nor is there any reason the question of whether even the useful arts should be fostered cannot be revisited. In an evolving world it may become undesirable to incentivize technol-

the Constitutional regime the Congressional patent power only extends to promotion of progress in the useful arts, the answer to when (if ever) software constitutes patentable subject matter is tied to when software claims should be viewed as falling within the targeted useful arts. That determination requires not only analysis of the reach of term standing alone, but because the exercise of the patent power is discretionary with Congress, whether that body has imposed additional limitations.

1. *The Useful Arts*

The term "useful arts" is not defined anywhere in the Constitution. However, the extensive literature contrasting the "sciences" with the "useful arts" for purposes of determining the respective scope of the copyright and patent laws under the Constitutional provision sheds some light.²⁰¹ That discussion indicates that protection of expression ("writings") of general knowledge or learning ("science" in 18th century terms) was placed exclusively under the copyright laws, and protection of applied science ("useful arts") exclusively under the patent laws.²⁰² Under this reading, the useful arts, and therefore the maximum Constitutional reach of United States patent law, are limited to the practical application of knowledge and learning to the industrial arts, that is to the creation of specific tools, products and procedures for performing activities such as the creation of new things, arrangements of things or relationships between things.²⁰³ The courts have consistently applied this interpreta-

ogy or at least some branches of technology. See *infra* note 238 and accompanying text. In this case, as the exercise of the patent power is not mandatory, these assessments could be dealt with directly by Congress. See discussion *infra* notes 230-233 and accompanying text. If an absolute bar were desired, however, a Constitutional amendment would be required to eliminate Congresses' power under the Patent Clause.

Both of these questions require substantial empirical input and are beyond the scope of this inquiry.

201. The commentators are virtually unanimous in parsing the clause to separate out the copyright and patent authorizations as follows: useful arts, inventors, and discoveries are the subject of patent laws and science, authors, and writings are the subject of copyright laws. See LIPSCOMB, *supra* note 34, § 2.1. See *infra* note 285 and accompanying text (discussing the proper roles of copyright and patent law in protecting software innovation).

202. See CHISUM, *supra* note 18, § 1.01; LIPSCOMB, *supra* note 34, § 2.1.

203. See CHISUM, *supra* note 18, § 1.01 ("The general purpose of the statutory classes of subject matter is to limit patent protection to the field of applied technology, what the United States Constitution calls the 'useful arts.'"); LIPSCOMB, *supra* note 34, § 2.1 ("the words 'useful arts' were intended to mean practical or industrial arts."); MERGES ET AL., *supra* note 70, at 151; *In re Alappat*, 33 F.3d 1526, 1551-52 (Fed. Cir. 1994) (Archer, C.J., dissenting). See also *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1373 (Fed. Cir. 1998) (applying a "useful, concrete and tangible" output test to find utility). Technology comes in two primary flavors under the patent laws: products and processes. The former are machines, articles of manufacture and compositions of matter, the later methods of performing tasks. The former are generally viewed as "technology" per se. It is required the applicant demonstrate the latter involve the technological arts. See CHISUM,

tion, with the CAFC expressly equating the Constitutional "useful arts" with the present day term "technological arts."²⁰⁴

Although this basic definition provides general guidance, it requires refinement for specific application. Much has been made in the patentable subject matter cases over the past 15 years of the basic Congressional premise cited by the Supreme Court in *Diamond v Chakrabarty*²⁰⁵ that the Patent Act is meant to "include anything under the sun that is made by man."²⁰⁶ If this were truly the case, the task would be a simple one: everything humans create are "useful arts" and should be patentable subject matter. Such a sweeping statement is clearly at odds with the above interpretation of the Patent Clause itself, with only the more limited class of technological activities falling within the useful arts. In fact, although the Court in *Chakrabarty* clearly instructed the lower courts to adhere to the guiding spirit of this pronouncement, it did go on to explain that there are a variety of human activities that do not merit the embarrassment of a patent on policy grounds.²⁰⁷ As always, to include some things is to exclude others. The difficulty comes in determining what is in and what is out. The core normative task in the software as patentable subject matter debate is determining on which side of the divide each particular software invention falls.

Judicial attempts to refine the "useful arts means technology" definition for application generally reflect one of two approaches: exclusion of entire classes of non-technological activities and exclusion of inventions deemed too inchoate to qualify as technology. Both have been largely unsuccessful in the software arena, but the reasons for each failure provide important guidance in making additional efforts.

The first approach seeks to identify and exclude software inventions when they implicate activities which fall outside the technological arts. The approach lead to the development of doctrines such as the un-

supra note 18, § 1.01. In the case of software which can function either as a language to describe a process or as the means for actual computer (technological) implementation of a process, the useful arts classification analysis can be extremely confusing. See discussion *supra* Part III.G.

204. See, e.g., *In re Musgrave*, 431 F.2d 882, 893 (C.C.P.A. 1970); *In re Waldbaum*, 457 F.2d 997, 1003-04 (C.C.P.A. 1972) (Rich, J., concurring); *Alappat*, 33 F.3d at 1552-53 (Archer, C.J., dissenting).

205. *Diamond v Chakrabarty*, 447 U.S. 303, 309 (1980).

206. S. REP. NO. 1979, 82d CONG. (1952); H.R. REP. NO. 1923, 82d CONG. (1952).

207. *Diamond*, 447 U.S. at 314-17. This sweeping statement was, at least in part, in response to the earlier Supreme Court software decisions in *Gottschalk v. Benson*, 409 U.S. 63 (1972) and *Parker v. Flook*, 437 U.S. 584 (1978), where the Court seemed to read the Patent Act narrowly, excluding what it regarded as inventive developments unanticipated by the Congress. *Id.* In *Chakrabarty* the Court was making clear that patent law by its very nature was dealing with the unanticipated (invention) and should be viewed broadly. See discussion *infra* notes 230-234 and accompanying text.

patentability of "methods of doing business" and "mental steps."²⁰⁸ Although at the conceptual level such an approach makes sense, it has been unworkable in practice. The fundamental difficulty is that even when the target application for a software invention is clearly a non-technological activity, it does not follow that the computer tool, itself falls outside the technological arts.²⁰⁹ Determining when a particular software claim is to the unpatentable activity or to the computer tool, using labels such as "method of doing business" or "mental steps," without more, has proven problematic at best.²¹⁰ Even if the specific results have been less than satisfactory, keeping the important "activity versus tool" difference in mind is helpful to full resolution of the questions surrounding the patentability of computer software.²¹¹

The second approach focuses on claims which are too inchoate to pass muster as technological application. They include the well-known doctrines of the unpatentability of laws of nature, natural phenomena and abstract ideas. This approach gave rise to the failed "mathematical algorithm" test in the software arena.²¹²

All three legs of the underlying framework pre-date the computer software patentability debate by many years.²¹³ The first two legs, denying patents for claims to laws of nature and physical phenomena standing alone, arose in connection with much earlier technology inventive forays, including the telegraph,²¹⁴ the telephone,²¹⁵ radio antennas²¹⁶ and pre-recombinant DNA "biological combination"²¹⁷ efforts. The rationale for these exclusions has been varyingly stated as: these matters

208. None of these doctrines are expressly set out in the Patent Act. They are predicated instead on the reviewing court's view that the substance of the claim does not involve activity within the "useful arts."

209. See discussion *infra* note 293 and accompanying text.

210. The doctrines are frequently applied as additional justification for otherwise difficult to articulate holdings of unpatentability. This eventually leads their abandonment in favor of a more direct analysis. See *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1373-74 (Fed. Cir. 1998). ("We take this opportunity to lay this ill-conceived exception [the method of doing business test] to rest."); *In re Musgrave*, 431 F.2d 890-91 (C.C.P.A. 1970) (abandoning the "mental steps" doctrine as a "morass"). Another example is the death of the "function of a machine" doctrine in *In re Tarczy-Hornoch*, 397 F.2d 856 (C.C.P.A. 1968).

211. See discussion *infra* notes 293-297 and accompanying text.

212. See *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1373-74 (Fed. Cir. 1998); *Gottschalk v. Benson*, 409 U.S. 63 (1972); *Parker v. Flook*, 437 U.S. 584 (1978); *Diamond v. Diehr*, 450 U.S. 175 (1981); see discussion *supra* notes 84-87 and accompanying text.

213. All of these categories are of judicial origin; none appearing in the Patent Act.

214. *O'Reilly v. Morse*, 56 U.S. (15 How.) 62 (1854).

215. *The Telephone Cases*, 126 U.S. 1 (1887).

216. *MacKay Radio & Telegraph Co. v. Radio Corporation of America*, 306 U.S. 86 (1939).

217. *Funk Bros. Seed Co. v. Kalo Inoculant Co.*, 333 U.S. 127 (1948).

are "part of the storehouse of knowledge of all men,"²¹⁸ they are discovered not invented²¹⁹ and "they are the basic building tools of scientific and technological work."²²⁰ These articulations reflect an underlying policy concern that permitting preemption of naturally occurring forces or relationships through patent law, thereby prohibiting their use in the entire range of potential technological applications, will severely constrain progress in the useful arts, rather than promote it.²²¹ Consequently, these claims do not merit the "embarrassment of an exclusive patent" and the subject matter is left "free to all men and reserved exclusively to none."²²²

The third leg of the exclusion, abstract ideas, also has venerable pre-computer lineage. It arose, however, in connection with a low-technology case involving the attachment of an eraser to the end of a pencil. In *Rubber-Tip Pencil v. Howard*,²²³ the Supreme Court found the claims sought merely to preempt the general concept without any "inventive" application.²²⁴ Stating that "[a]n idea of itself is not patentable"²²⁵ the

218. *Id.* at 130.

219. See *Parker v. Flook*, 437 U.S. at 584, 591-92; *Funk Brothers*, 333 U.S. at 130. Ironically, the language of the Constitution actually speaks in terms of "discoveries" rather than "invention." The Court has redefined the former as the latter for purposes of the Patent Clause. See *Funk Brothers*, 333 U.S. at 132 ("There is no way in which we could call it such unless we borrowed invention from the discovery of the natural principle itself. [. . .] But we cannot so hold without allowing a patent to issue on one of the ancient secrets of nature now disclosed.")

220. *Gottschalk v. Benson*, 409 U.S. 63, 67 (1972).

221. One could view the claimant to a law of nature as a human technological artifact rather than a discovery of some Platonic essence; the creation of an imperfect but operable description of naturally occurring things or relationships captured in a technological "working model." The main difficulty, as noted in the text, is that there may be disproportionate costs in foreclosing general access to these general "building block" principles, particularly if scientific inquiry is already sufficiently incentivized through other means.

222. *Funk Brothers*, 333 U.S. at 130.

223. *Rubber-Tip Pencil v. Howard*, 87 U.S. (20 Wall.) 498 (1874).

224. Although *Rubber-Tip* is frequently cited for the non-patentability of "abstract ideas" arguably the holding is actually related to non-obviousness concerns. The applicant in that case set out the idea of attaching a rubber eraser to the end of a pencil by making a cavity in the eraser. The invention clearly falls within the useful arts (pencil erasers). The application clearly demonstrates how the idea is to be implemented (the cavity in the eraser). The actual holding in the case reads as follows:

Everybody knew, when the patent was applied for, that if a solid substance was inserted into a cavity in a piece of rubber smaller than itself, the rubber would cling to it. . . . An idea of itself is not patentable, but a new device by which it may be made practically useful is. The idea of this patentee was a good one, but his device to give it effect, though useful *was not new*. Consequently he took nothing by his patent.

Id. at 507 (emphasis added).

This language reads like the equivalent of a modern § 103 obviousness rejection (§ 103 being adopted subsequently). It seems, therefore, that *Rubber Tip* merely contains dicta applicable to the inability to claim a mere idea without specific application to the technolog-

Court held the patent invalid. In essence, a "great thought," even if its ultimate application falls within the technological arts once implementation issues are resolved, does not sufficiently further the specific patent law objective of technological advance. Quite the contrary, prematurely granting a patent in such cases severely limits inventive behavior aimed at achieving practical technological implementation of the idea. Abstract ideas, without more, are not yet useful enough arts to merit a patent.

Not surprisingly, the touchstone for patentability of inventions which raise concerns in all three of these areas ties back to the Constitutional useful arts requirement. Although laws of nature and natural phenomena are not themselves patentable, "a novel and useful structure created with the aid of knowledge of scientific truth may be."²²⁶ Likewise, in the realm of abstract ideas, although "[a]n idea is not patentable, . . . a new device by which it may be made practically useful is."²²⁷ These tests reflect the patent law's focus on technology. Entitlement to a patent requires that the details for one or more specific implementations of the law of nature, natural phenomena or general idea be sufficiently identified and fleshed out before it addresses patentable subject matter. The patent is then issued on and restricted to that specific implementation.

Unfortunately, as experience with the mathematical algorithm permutation of the "law of nature/abstract idea" test has demonstrated, something other than the mathematical algorithm test is required in the software context.²²⁸ The rationale giving rise to the concepts, however, provides useful policy guidance. The same underlying concerns driving a desire to avoid patents on general statements of laws of nature and abstract ideas apply equally in the context of software claims. A workable test must consistently separate software claims to matters falling within the too technologically inchoate classes of laws of nature, natural phenomena and abstract ideas from those which involve their application, the useful arts.²²⁹

ical arts. Another expression of the abstract idea concern is found in Justice Frankfurter's concerning opinion in *Funk Brothers*, 333 U.S. at 134. See *infra* note 229 and accompanying text.

225. *Rubber-Tip*, 87 U.S. at 507.

226. *MacKay Radio and Telegraph Co. v. Radio Corp. of Am.*, 306 U.S. 86, 94 (1939). See also *Diamond v. Diehr*, 450 U.S. 175, 187-88 (1981) (a good review); *Gottschalk v. Benson*, 409 U.S. at 67; *Funk Brothers*, 333 U.S. at 130.

227. *Rubber-Tip*, 87 U.S. at 507.

228. See discussion *supra* notes 84-87 and accompanying text.

229. The abstract idea/application dyad is the source of the *Guidelines* "utility" formulation which seeks to avoid the mathematical algorithm difficulties. See *supra* note 71 and accompanying text. The CAFC's recent decision in *State Street Bank* supports this approach stating: "The question of whether a claim encompasses statutory subject matter should not focus on which of the four categories of subject matter a claim is directed to—

process, machine, manufacture, or composition of matter—but rather on the essential characteristics of the subject matter, in particular, its practical utility.” *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1375 (Fed. Cir. 1998). To make these determinations the court equates unpatentable mathematical algorithms with abstract ideas. As the court notes: “[T]his means that to be patentable an algorithm must be applied in a ‘useful’ way.” *Id.* at 1373. The measure of this requirement, and therefore of patentable subject matter question, is whether the idea has been “reduced to some type of practical application, i.e., ‘a useful, concrete and tangible result.’” *Id.* This is entirely consistent with the Supreme Court’s abstract idea analysis. See discussion *supra* note 227 and accompanying text. The problem is that the court’s actual application and analysis in *State Street Bank* fails to distinguish between apparatus and process claims.

The court found, properly, that the invention actually claimed in *State Street Bank* involved a statutory machine for managing the financial matters of the particular mutual fund partnership arrangement. *Id.* at 1371-72. In such cases (apparatus claims) the reason the mathematical algorithm/abstract idea issue disappears and there is no need for the FWA test is that the mathematical algorithm/abstract idea for making the allocations is incorporated into and applied by the particular claimed machine. The machine itself, a tool for performing the targeted transformation, is the necessary “useful, concrete and tangible result” within the useful arts. See discussion *infra* note 293 and accompanying text.

Assume instead that claims covered the *process* for performing the asset and cost to final share price transformation or even more generally to the underlying allocation algorithm itself. A patentable subject matter test must still be applied. In this situation, however, there is no apparatus application of the mathematical algorithm/abstract idea. Some other useful arts application must be found. It may be appropriate to test the process outputs as one method of determining if the process application falls within the useful arts. See *infra* note 297 and accompanying text. But this is a different inquiry than when apparatus, which automatically satisfies the patentable subject matter requirement, is involved.

The court appears to confuse the two situations in its *State Street Bank* holding:

[T]he transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm, formula, or calculation, *because it produces ‘a useful, concrete and tangible result’—a final share price momentarily fixed for recording and reporting purposes . . .*

State Street Bank, 149 F.3d at 1373 (emphasis added).

Under the facts of *State Street Bank*, which involves claims to a machine, § 101 is not satisfied because the final share price output of the machine is a “useful, concrete and tangible result.” The machine implementation itself satisfies § 101; the machine which performs the transformation is the application of the abstract idea. The nature of the particular transformation performed by the machine or of the output is irrelevant. Assessing them, as the court appears to, confuses claims to apparatus with its field of application. See discussion *supra* notes 209-211 and accompanying text; see *infra* notes 293-297 and accompanying text. The same difficulty appears in the court’s analysis of the justifications for *Alappat* and *Arrhythmia Research*. See *State Street Bank*, 149 F.3d at 1373.

Finally, the court’s “utility” test— “[u]npatentable mathematical algorithms are identified by showing they are merely abstract ideas constituting disembodied concepts or truths that are not ‘useful’” (*Id.* at 1373)—confuses the limitation of patentable subject matter to the useful arts with the separate requirement of usefulness. The patentable subject matter only addresses whether the claimed invention falls within the target useful arts as defined by the four categories listed in § 101. The usefulness requirement exists to ensure that the invention does what is claimed, generally whatever that may be: a smooth waveform, the condition of a patient’s heart or the final share price. See discussion *supra* notes 69-70 and accompanying text. Claims to a machine which attempts to do those things targets inven-

2. Congressional Action

As noted above, because the grant of the patent power is not mandatory, merely recognizing that the permissible scope of congressional action is limited to the technological arts (poorly defined as that may be) does not end the policy inquiry. Congress retains the power to include or exclude any or all specific "useful arts" inventions based on its case-by-case evaluation. It is, therefore, important to determine if Congress has made specific determinations which might affect the assessment of software as useful arts patentable subject matter.

The Congressional view on the question is expressed in § 101. The Supreme Court had the opportunity to interpret this provision and, in particular, the term "manufacture" which is especially relevant to the present inquiry, in *Chakrabarty*. Noting the now frequently quoted language from the Congressional Report accompanying the 1952 Patent Act indicating that statutory subject matter should "include anything under the sun that is made by man," the Court found that "[t]he subject matter provisions of the patent law have been cast in broad terms to fulfill the constitutional and statutory goal of promoting 'the Progress of Science

tion in the useful arts. If the resulting machine does not succeed it is not useful and no patent will issue (there may also be an enablement problem under § 112). By referring to usefulness, rather than useful arts, as the measure of § 101 compliance the court appears to inappropriately add another dimension to the patentable subject matter inquiry; in particular that the output must also have certain characteristics.

The court is correct that the § 101 test is whether an abstract idea has been applied. Because machines are technology, they should be treated as satisfying the § 101 requirement regardless of the test applied. One could argue that all machine outputs will, in practice, satisfy the "useful, concrete and tangible" test so no issue will arise. On the face of *State Street Bank* that assertion is troublesome; how would *Schrader*, *Maucorps* and *Meyer* be resolved if they had each claimed special purpose computers implementing their respective processes? Cf. *State Street Bank*, 149 F.3d at 1373-74 (seeming to indicate unpatentability of the process in all three cases but noting they predate the new test). The most reliable approach is to directly determine whether the claim is to apparatus (patentable subject matter) or a process (which must still be tested). The implementation-language test proposed in this article solves this issue by separating apparatus and process software claims, treating the former as patentable subject matter and permitting the latter to be tested under generally applicable process tests without confusion arising from their software expression. See *infra* Part III.G (in particular notes 302-311 and accompanying text, discussing how it deals with the abstract idea concern).

Justice Frankfurter, concurring in *Funk Brothers*, stated he believed that labels like "laws of nature" only served to confuse the issue because such terms are "vague and malleable terms infected with too much ambiguity and equivocation. Everything that happens may be deemed 'the work of nature,' and any patentable composite exemplifies in its properties 'the laws of nature.'" *Funk Brothers*, 333 U.S. at 135. To avoid using another label, Justice Frankfurter analyzed that case in terms which appear to apply a § 112 enablement test. Although this would resolve the particular case, something more is required to draw the abstract idea versus technology distinction for purposes of § 101. See discussion *infra* notes 302-311 and accompanying text.

and the useful Arts' with all that means for the social and economic benefits envisioned by Jefferson."²³⁰ The Court went on to admonish against the addition of any limitations not expressly stated under § 101; that is beyond requiring the invention be a machine, article of manufacture, process or composition of matter.²³¹

Of additional relevance are the Court's observations concerning inventions "in areas not contemplated by Congress when the patent laws were enacted."²³² Addressing the concern over the potentially adverse consequences of recombinant-DNA technologies to society, the Court noted that "[a] rule that unanticipated inventions are without protection would conflict with the core concept of the patent law that anticipation undermines patentability" and "that Congress employed broad general language in drafting § 101 precisely because such inventions are often unforeseeable."²³³

230. *Diamond v. Chakrabarty*, 447 U.S. 303, 315 (1980).

231. *Id.* at 308. See also *State Street Bank*, 149 F.3d at 1373 ("The repetitive use of the expansive term "any" in § 101 shows Congress's intent not to place any restrictions on the subject matter for which a patent may be obtained beyond those specifically recited in § 101.").

232. *Chakrabarty*, 447 U.S. at 315. ("Flook did not announce a new principle that inventions in areas not contemplated by Congress when the patent laws were enacted are unpatentable per se.")

233. *Id.* at 316. Not only does this language provide interpretative guidance regarding the meaning of § 101, it provides good policy guidance for Congressional exercise of its exclusionary power under the patent laws. Because the patent laws are aimed at invention, with its attendant unpredictability, careful consideration should be given to any exclusions which may limit the scope of basic inventive behavior. The core problem is there will frequently be no way to effectively determine at the pre-invention stage which technologies to remove from the system. It is possible to recognize and define many, if not most, problems only in the course of development, implementation or use. Therefore, assuming we generally wish to foster technical advance, it is best generally to give free rein to the direction of inventive impulses and focus instead on controlling application of the results as specific difficulties emerge. This approach also recognizes that because research and technology are driven by more than the patent laws, it is unlikely merely removing or limiting the patent incentive will provide effective deterrence to the development of undesirable technologies. See, e.g., *id.* at 317. If continued research or other technological activity is deemed to be undesirable and, therefore limited or prohibited, such direct regulatory prohibition will be more effective than removal of the patent incentive.

The direct post-invention regulatory approach also has the benefit of allowing the patent incentive to adjust automatically as the level of regulation changes, eliminating the need for constant tinkering with the patent statutes. Even though regulated or prohibited activities would still be technically included within the patent incentive, the regulation would act as an appropriate adjustment to the degree of incentive, reflecting the amount and form of the regulation.

To the extent we only engage in limited policy fine-tuning of intellectual property incentives, we should recognize and accept that intellectual property laws will act a bit like legal "black-holes" trapping anything that comes within its event horizon. Justice Scalia's comment in a recent copyright "first sale" oral argument that "we're talking about shampoo here" reflects precisely this lack of extensive precise policy making in the copyright laws.

In *Chakrabarty* the Court found that beyond the express language of § 101 itself (which should be broadly read even in the face of the unanticipated) Congress has done almost nothing on the exclusionary front.²³⁴ What the Court has to say about the congressional intent underlying the Patent Act is of binding effect.²³⁵ If Congress wishes to correct the Court's interpretation, it is required to act. Congress has infrequently responded to arguments in favor of excluding particular technologies from coverage under the patent laws, choosing instead to regulate risks on application.²³⁶ In the case of patent coverage of computer software Congress has consistently taken no action,²³⁷ leaving the matter entirely to evolving court interpretation in light of the Constitutional useful arts mandate.

The above demonstrates that the primary policy objective, as well as the primary limitation guiding the patentable subject matter inquiry, remains the Constitutional purpose of progress of useful arts as that concept has been illuminated by previous efforts of the courts and the commentators. What remains²³⁸ is to determine whether current

Of course, as experience in the operation of the patent (or other intellectual property) incentive structure reveals the need for adjustments, they can be made based on the identified facts and circumstances. *See, e.g.*, 35 U.S.C. § 271(e) (1994) (permitting certain pre-expiration uses of patented inventions in the r-DNA field in connection with obtaining FDA approvals).

234. The one exception relates to nuclear material. *See* 42 U.S.C. § 2181(a) (1994) (noted in *Chakrabarty*, 447 U.S. at 318).

235. *Chakrabarty*, 447 U.S. at 315 (citing *Marbury v. Madison*, 1 Cranch 137, 177 (1803)). It is always interesting to observe the Court's performance of this interpretive exercise. The computer software/patent law question is no exception. The Court has moved from a position which implies that if software is to be covered by the patent laws that it lies within the province of Congress to say so. *See* *Gottschalk v. Benson*, 409 U.S. 63, 72-73 (1972); *Parker v. Flook*, 437 U.S. 584, 595-96 (1978) to the opposite extreme that no implied restrictions on coverage should be added by the Court and it is up to Congress to carve it out. *Diamond v. Diehr*, 450 U.S. 175, 182 (1981).

236. *See supra* note 234 and accompanying text. Congress has occasionally expressly broadened the patent law reach when ambiguities appeared. *See, e.g.*, Plant Protection Act of 1930, 35 U.S.C. § 161 (1994) and the Plant Variety Protection Act of 1970, 7 U.S.C. § 2402 (1994). Both are discussed in *Chakrabarty*, 447 U.S. at 310-14.

237. To the extent software is technology, eliminating it from coverage under the patent laws may violate the GATT/TRIPS accord. *See* Final Act Embodying the Results of the Uruguay round of Multilateral Trade Negotiations and Marrakech Agreement Establishing the World Trade Organization, signed at Marrakech (Morocco), April 15, 1994, Annex 1C, Agreement on Trade-Related Aspects of Intellectual Property Rights, § 27(1). 33 I.L.M. 81.

238. Simply because software does not currently receive distinctive treatment does not mean as a normative policy matter we should not inquire whether it should. Assuming that the thesis of this article is accepted and software as implementation is entitled to the maximum patent incentive afforded to other technologies, the "software is special" inquiry would focus on reasons for limitation or exclusion. This argument can be made on two levels. The first is that software in general is bad for society. At present this concern does not appear to be motivating arguments against the patentability of software. If it were to

software as an article of manufacture practice is consistent with this pol-

be presented, it could take two forms. The first would reflect a view that all technology is bad and therefore the underlying Constitutional incentive to the "useful arts" is a fundamental mistake. This approach has nothing to do with software patentability as such. It implicates general social views regarding the desirability of technology as a whole. The patentability of software would stand or fall depending on the outcome of that larger policy debate, which would focus on the desirability of having a patent incentive for technology at all. The second, purer form would argue there is something particularly undesirable about software (or computing) in much the same way recombinant DNA research was targeted in *Chakrabarty*. It seems more appropriate to deal directly with these concerns through specific regulation aimed at mitigating or avoiding the particular problems of concern in lieu of an ineffective fine-tuning the patent law subject matter test in the vain hope that will create sufficient disincentives. See discussion *supra* note 233 and accompanying text.

The second level of the "software should be excluded" argument addresses whether the underlying economic incentive theory of the patent laws functions inappropriately for reasons unique to software or the software industry. The question has been the subject a long debate, on-going through the present. There are arguments from all sides, ranging from support of the current regime, through a variety of *su generis* protection proposals (enhancing or reducing protection) to concerns that the software industry will be crushed by the weight of software patents and so no protection should be provided. There is even the anthropomorphic position that "software wants to be free." The debate has consumed many pages in many publications. See Chisum, *supra* note 18, 1997 Supplement at 23 n.191 (citing numerous articles). There have been several sets of extensive set of hearings, including one by CONTU and a 1994 Congressional review with members of the software industry. See STOBBS, *supra* note 18, §§ 1.13-1.15. Although the results remain inconclusive, the issue is critical to determination of the appropriate policy position. See discussion *supra* notes 194-198 and accompanying text.

Because until recently software patents have been relatively difficult to obtain, the primary risks thus far have been of those of omission. As a result some practical comfort has been taken from the fact that the software industry continued to prosper. See, e.g., *Diamond v. Diehr*, 450 U.S. 175, 217 n.43 (1981) (Stevens, J., dissenting). This comfort does not, however, rest on any assurance that the patentability line was appropriately drawn as an economic matter, but merely on the observation that not much harm appears to be arising from any omitted incentive. An economic downturn in the industry could cause us to become less sanguine. More immediately, the loosening of the patentability requirements under the *Guidelines* may shift the focus to problems of commission. With increased patent issuance the software industry is beginning to feel the related economic costs of increased assertions, disputes and licensing. See, e.g., *Big Blue is Out to Collar Software Scofflaws*, BUS. WEEK 39 (Mar. 17, 1997). This will no doubt bring calls for scaling back the patent incentive for "the good of the industry."

In either situation (over- or under-incentive), it is essential that improper operation of the existing system does not distort our analysis. For example, if increasing numbers of software patents begin to cause havoc in the industry, we need to understand the real source of the difficulty. The problem may stem from the inherent ability of the software industry to accommodate the level of incentive offered by the proper operation of the current patent system. But it may actually arise, as this article argues, from issuance of unjustified software patents as a result of improper over-inclusion. Depending on the cause, different solutions will be required. By identifying and resolving the issues arising from improper operation of the present system, the language versus technology dichotomy proposed in this article helps ensure that we only debate this second "software is special" question in connection with the proper functioning of the existing patent system.

icy objective and if it is not, as this article argues is the case, why not and how appropriate adjustments can be made.

3. *Can the Current "Software as Article of Manufacture" Practice Get Us There?*

a. *Bounding the Inquiry*

To profitably pursue the question of whether software claimed in article of manufacture form should be viewed as within "technological arts," it is useful first to bound the target of the inquiry.²³⁹ This requires a brief discussion of the scope of the article of manufacture form of claim under consideration and the nature of software.

b. *Scope of the Claim Format*

First, clearly "pure" article of manufacture claims covering only inventions relating to the hardware carrier itself, *e.g.*, the structure or composition of the diskette, ROM, CD or the like, are not controversial. There can be little debate that content free claims to the computer readable media alone constitute § 101 articles of manufacture.²⁴⁰ These claims are of interest, however, as they provide an intuitive, although incorrect, basis for the software as article of manufacture argument, because the hardware carrier standing alone is an article of manufacture, perhaps such a carrier *plus* its software content should also be considered an integrated article of manufacture.

Second, article of manufacture claims involving data content are outside the discussion. Although both data and software program content are (or can be) machine readable, they serve very different functions and raise very different patentability issues. Computer software consists of a set of instructions that cause a computer system to perform a particular sequence of operations directed toward accomplishing the particular task defined by the program. Data is the informational raw material on which the programmed computer system operates.²⁴¹ Although the data affects the specific outcome of the computer operation in terms of result and possibly the order in which particular operations are performed, it does not create the operational framework for the implementing com-

239. Software can be claimed in a variety of formats, including specifically as process claims. This discussion focuses on the "article of manufacture" format: those claims involving media as a carrier for computer program instruction. However, the test developed in this article for making the useful arts determination has generally applicability to software claims.

240. Although the purpose of the device is to hold software information in computer readable form, the claims themselves cover only the physical media-information interactions relating to storage, modification and retrievability. They do not implicate the content, organization or functionality of the information or the use of the informational content.

241. Examples include music, images, numerical information and text.

puter system. Although ambiguous "data content" article of manufacture claims can raise § 101 issues, they are readily resolvable against their patentability during the examination process.²⁴² This article,

242. There is no debate that a claim to pure data standing alone is not patentable subject matter. The concern, as expressed by the PTO Board majority in *Beauregard* and the dissent in *Alappat*, is that because data and software instruction are technically indistinguishable on the carrier (both are electronic encoding), if software content claims are treated as patentable no valid line can be drawn preventing data content from also receiving patent protection. See *PTO Board, supra* note 156, at 15-16; *In re Alappat*, 33 F.3d 1526, 1566 (Fed. Cir. 1994) (Archer, C.J., dissenting). There is some validity to this concern based on a narrow reading of the apparent reliance in *Lowry* on the *Bernhart* physical electronic structure approach to find patentable subject matter. See, *In re Lowry*, 32 F.3d 1579, 1583-84 (Fed. Cir. 1994).

In the examination context, however, this concern is a non-issue. The fact that both data and program content appear as electronic coding on the medium does mean they each physically alter the medium and that a program can be played as "music" or music could be loaded and "run" as a program. Neither has any bearing on the actual application and examination process. The applicant does not provide the examiner with a computer readable medium with electronic coding and ask for a patent. The application must describe and distinctly claim the invention. 35 U.S.C. § 112 (1994). The applicant must, therefore, "fess up" in the claims as to whether the content is being claimed as data or program content. If the electronic coding is claimed as data, regardless of whether it is data or program content in fact, there is no § 101 useful art involved and no patentable subject matter. If an applicant claims data content as software instruction, the invention will fail even a weak test of usefulness; the instructed computer system does nothing beyond the circular "execute the program content." See discussion *supra* notes 69-70 and accompanying text. This is precisely the line the *Guidelines* found with relatively little effort in the media context by using in the functional versus non-functional descriptive material distinction. See *Guidelines, supra* note 4, at 7481-82 (discussing the distinction between the data constituting the song and the software for playing the music); see discussion *supra* notes 170-176 and accompanying text.

There is a good argument that putting data (or software) on computer readable media (or anything else tangible for that matter) makes the combination patentable subject matter. That does not, however, make the combination patentable. See discussion *supra* note 240 and accompanying text; see *infra* note 274 and accompanying text; R. Carl Moy, *Statutory Subject Matter and Hybrid Claiming*, 17 J. Marshall J. Computer & Info. L. 277 (1998). (providing a method for dealing with such hybrid inventions).

The data versus program physical encoding and the carrier combination confusion provide further support for abandoning the "hardware" carrier approach used in the *Guidelines*. By permitting claims to software "as such" without media, the implementation-language test proposed in this article relegates the confusion arising from even indirect reliance on the physicality of electronic coding as the basis for patentability to the irrelevance it deserves. The example of a CD loaded with a new song used by the PTO Board in *Beauregard* and the dissent in *Alappat* demonstrates the difference. Because under the implementation-language approach the software patentability analysis does not rely on either computer readable media or electronic encoding the fact that the music data is on the same type of computer readable medium as may carry software or stored in similar electronic form is simply irrelevant. All that matters is whether the claimed invention is limited to and can serve as a component of a computer system. See *infra* notes 259-277 and accompanying text.

therefore, focuses only on the more problematic software content article of manufacture claims.

c. Understanding Software: Technology, Language or Both?

As noted above, software consists of a set of instructions directing the operation of a computer system. When the software instructions are executed by the computer system, the system performs the specific task defined by the software program. It has been frequently and properly observed that software operates algorithmically. The program's instruction set is based on and reflects a particular methodology for accomplishing the defined task.²⁴³ A person can review the software code and understand this incorporated algorithm. Consequently, software extends beyond mere computer system instruction. It can also serve as a means to express and communicate the incorporated methodology.

The two functions of software produce very different outcomes to the software as patentable subject matter inquiry when passed through the patent policy test developed above. When used to communicate the contained algorithm to a person, software serves precisely the same function as any other language or means of expression. It involves the symbolic representation of the incorporated methodology in a form which permits its interpretation by human readers. There is nothing, therefore, which makes couching claims in software terms for this purpose technological or deserving of patent protection. In contrast, when software acts as computer system instruction the software serves as the actual means for implementing that same methodology, causing and directing the operation of the computer system's performance of the specific task. This function is technological and comes squarely within the useful arts targeted

The question as to whether claims to a data structure falls within § 101 is more interesting. This can be best resolved by first differentiating claims to data and claims to software implementation. If the claim relies on the specific data content it is non-statutory. If the claim covers only the organizational structure for data, independent of the particular data content, then it can be treated as any other software claim. The software claims are then tested as any other software claim under the proposed implementation-language dichotomy test. The data structure issue raised by *In re Warmerdam*, 33 F.3d 1354 (Fed. Cir. 1994) and *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994), does not concern the absence and presence of media, but whether the claims are really to data or software implementation, and if the latter, whether they restrict the use of the software exclusively to use within a computer system. Assuming *Warmerdam* avoids the data claiming issue (see *supra* note 154 and accompanying text) the difference in the cases is apparent: In *Lowry* the claims are limited to computer application, whereas in *Warmerdam* they are not. See discussion *infra* notes 254-258 and accompanying text.

243. There is no need for the steps of the algorithm to be executed sequentially. For example, the software can direct execution in parallel by multiple processors. The code will, however, always set out the steps of a process to be performed by the system.

by the patent laws. The objective, therefore, is to find a reliable way to consistently separate the former use of software from the latter.

In "real world" implementations, software programs can be expressed in a variety of computer programming languages. As the phrase "software code" implies, these computer languages vary in their suitability as vehicles for human communication or for computer instruction. For example, a program written in the COBOL programming language²⁴⁴ has a readily ascertainable connection to human language expression. An assembly language version of the same computer program is substantially more cryptic and will communicate very little to many readers. And the program expressed in machine code, consisting of the string of 1s and 0s commonly associated with computing, will communicate virtually nothing to anyone but a computer expert (and then only with considerable effort). Similarly, a computer system will require varying degrees of "translation" of each of these languages to effect the actual electronic operation of the hardware, more for higher level languages like COBOL, less for machine code. Generally stated, the less comprehensible software is to a person the more directly useable it is by a computer. Consequently, there is at least a superficial appearance of separation between the communicative and implementation uses of software based on the level of human readability versus computer usability of the particular software expression.

In the context of patent claims, however, even this glimmer of distinction frequently disappears and the computer instruction and human communication functionality of software become completely intertwined. Patent applicants frequently eschew direct claims to specific software program code as irrelevant to the invention. Instead they frame claims as a series of "means for" operational steps describing the incorporated methodology of the program instructions leaving any listing of particular program code exclusively to the detailed description in the specification.²⁴⁵ The result is an apparently complete merger of the two functions of software: the claims simultaneously serve as a means for expressing the contained algorithm to the reader in normal, if specialized patent claim, English language and simultaneously obtain patent coverage for the actual technological means for instructing the computer

244. The COBOL programming language is something of a relic from the early days of computer technology. With the year 2000 problem upon us, implicating large numbers of old COBOL programs, there is some frantic dusting off of these old programming skills.

245. See *supra* note 66 and accompanying text (discussing means for claim formats). For purposes of this discussion it is irrelevant whether the specification contains specific code, block diagrams, flowcharts or indicates that one skilled in computer programming can easily convert the claims narrative into one or more specific implementations. The latter format may, of course, have ramifications concerning novelty and non-obviousness. See discussion *supra* notes 47-48 and accompanying text.

system's operation.²⁴⁶

The ability to properly determine when software inventions should be classified as useful arts technology depends on the ability to reconstruct a dividing line between software as a language (not technology) and software as computer implementation (technology) in the patent claims context.²⁴⁷ A proper test for patentability of software related inventions must clearly and consistently draw a line separating claims to software as the specific means for computer system implementation of the contained algorithms/processes (which are patentable subject matter) from those using a software context merely to express and communicate those algorithms/processes (which must be tested on their own merit independently of the software context to determine if they involve patentable subject matter).

F. SOFTWARE UNDER THE GUIDELINES

Having identified the dual functionality of software and the need to distinguish these functions in the context of patent claims, we can turn to an assessment of the viability of the current software as article of manufacture practice set out in the Guidelines. As will be demonstrated, it is not up to the task.

It is important to recall that the Guidelines are based on the Commissioner's concession described in *Beauregard* that computer software on computer readable media is patentable subject matter.²⁴⁸ The PTO Board majority opinion in that case provides important insight concerning the thinking behind the PTO's concession and the resulting positions taken in the Guidelines. The PTO Board majority determined the claims were to an article of manufacture consisting of the computer readable medium and the carried software program.²⁴⁹ Applying a printed matter test and finding no functional relationship between the printed matter (the computer program) and the substrate (the hardware carrier), the PTO Board sustained the examiner's rejection under § 101.²⁵⁰ Clearly prior to the CAFC dismissing the "printed matter" analysis in *Lowry*, the PTO Board majority did not believe that the presence of the computer readable medium added anything which made the claims to the software

246. This "merger" of expression and function also raises concerns that there will be unjustified or improper overlapping copyright and patent protection for the same program. See discussion *infra* note 285 and accompanying text.

247. The language versus implementation dichotomy proposed in this article should not be confused with the expression-idea dichotomy that has filled so many pages in connection with copyright protection of computer software. They do not address the same issue. See *infra* note 285 and accompanying text.

248. See discussion *supra* notes 156-160 and accompanying text.

249. See *PTO Board*, *supra* note 156, at 5.

250. See *id.* at 5-7.

itself patentable subject matter. In fact, under the PTO Board majority's printed matter analysis, including the hardware actually clarified the case against patentability of the software as the applicant's use of the computer readable medium to obtain article of manufacture classification specifically prohibited any reliance on the software content.²⁵¹

Interestingly, the PTO Board majority opinion contains language reflecting, although not completely articulating, the need to distinguish between the implementation and language roles of software. Rejecting an argument that the software and diskette combination formed an article of manufacture machine component, the majority stated:

We do not agree that because appellant's article of manufacture is intended for use in a computer, or because its primary purpose is to control a machine, it is somehow transformed into a machine component. Similarly, we do not agree that since this program is intended to be read only by a computer, not a human, it is no longer printed matter but is somehow beyond printed matter. In the first place, appellants' list of instructions can be read by a human as fully discussed above [referring to the examiner's argument that "computer readable media" includes a sheet of paper]. Secondly, none of the claims at issue recites a machine control. To the contrary, the claims are directed to a list of instructions which have only the potential to control a machine. . . . Under the present state of the law, printed matter is printed matter no matter in what language it is printed, and is not somehow transformed into something else because it can also be read and used by a computer.²⁵²

And further that:

There is only a prospective functional relationship which will occur only when the printed substrate (the diskette with the program) is entered into the machine. This functional interaction is what the claims of the recently issued [process] patent define, but such a definition is not present in these [article of manufacture] claims.²⁵³

The majority, therefore, all but expressly acknowledged the view expressed in *Alappat* that software, when introduced into a computer system, makes the general purpose hardware into a special purpose computer instructed by that software and that this functionality is technology. Their quite proper concern was that, as claimed by Beauregard, the realization of that useful arts software functionality was only "poten-

251. See *PTO Board, supra* note 156, at 24. This view was undoubtedly driven at least in part by the concern over the ability to draw a dividing line between media containing data, such as music, and software programs. *Id.* at 16. See *supra* note 242 and accompanying text. In contrast, the PTO Board minority, focusing on the "intended environment" for use of the computer program disk rather than a diskette with information written on it, was able to envision the claimed invention as an article of manufacture that controls the operation of a machine. *PTO Board, supra* note 156, at 31-32.

252. See *id.* at 10-11.

253. See *id.* at 12.

tial” rather than required. As a result the structure of *Beauregard*’s claims failed to move beyond the mere expression of the same algorithm already covered by the corresponding process patents. The majority had walked up to, but failed to specifically recognize, they were addressing the important division between software as language and software as implementation.

The Commissioner’s concession “that computer software programs embodied in a tangible medium such as floppy diskettes, are patentable subject matter under [§ 101],”²⁵⁴ however, completely failed to address the fundamental language versus implementation concerns raised by the PTO Board majority in *Beauregard*. Instead it placed full reliance on the presence of the “tangible medium” as the indicator of patentable subject matter.

In doing so, the Commissioner’s concession over-reads *Warmerdam* and *Lowry*. As discussed above, the two cases superficially appear to reflect the need for physical media in the claims, precisely as called for in the Commissioner’s concession.²⁵⁵ What actually explains the divergent outcomes, however, is a more fundamental difference in the two claim formats when analyzed against the language-implementation dichotomy. In *Lowry* the “potential” for machine control is transformed exclusively into a software as implementation claim, by the claim preamble stating the invention is “for access by an application program being executed on a data processing system.”²⁵⁶ In contrast, *Warmerdam*’s claim to “a data structure generated by the method of Claims 1 through 4” leaves any possible machine control merely as “potential,” there being no requirement that the invention actually be used as in a computer implementation.²⁵⁷ The different outcomes in the cases, therefore, actually reflect the fact that when analyzed as a whole the particular claims fall on different sides of the language versus implementation dichotomy, not a requirement that tangible media be present in the claims as reflected in

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

255. See discussion *supra* notes 152-155 and accompanying text.

256. The strong implication in *Lowry* that the claim preamble in that case should be read as requiring use of the data structure by a computer is consistent with a similar reading of preamble limitations in *Alappat*. See *In re Lowry*, 32 F.3d 1579, 1583 (Fed. Cir. 1994); *In re Alappat*, 33 F.3d 1526, 1544 (Fed. Cir. 1994). The *Guidelines* reject this position. See *Guidelines*, *supra* note 4, at 7483-84. Absent a preamble, although the presence of the computer readable medium carries a strong inference that such a limitation is part of the claims, as is discussed below, that is not sufficient to make the jump from software as language to software as implementation. See discussion *infra* notes 258-268 and accompanying text.

257. This assumes the data claiming issue is avoided. See *supra* note 154 and accompanying text.

the Commissioner's concession.²⁵⁸

As a result of this over-reading, the PTO position reflected in the Guidelines calls for examination of software claims based on the following two inter-related and faulty syllogisms premised on the importance of the presence or absence of computer readable media:

1. *Syllogism One*

To be a § 101 article of manufacture (patentable subject matter) the functionality of the software must be electronically implementable by a computer. Claims to software on computer readable media permit implementation of the software functionality. Therefore, claims to software on media are patentable subject matter as articles of manufacture.²⁵⁹

2. *Syllogism Two*

To be a § 101 article of manufacture (patentable subject matter) the functionality of the software must be electronically implementable by a computer. Claims to software that do not include computer readable media do not permit implementation of the software functionality. Therefore, software claims which do not include computer readable media are not patentable subject matter as articles of manufacture.²⁶⁰

Mercifully, the Guidelines clearly shift focus away from the use of mathematical algorithm forms of analysis, including the related *Freeman-Walter-Abele* two-step test. The computer readable medium test, however, fails to draw an appropriate line between software used as language and software claimed as implementation. As a result, the Guideline syllogisms are at once over-inclusive (the first syllogism) and under-inclusive (the second) from a policy perspective. An analysis of each syllogism demonstrates the difficulties.

3. *Syllogism One (Over-inclusion)*

The vice of over-inclusion is the possible application of patent incentive to inventions which as a policy matter we do not wish to encourage. Under the present Constitutional regime this includes all fields of human endeavor that do not involve technology: subject matter falling

258. It is hard to be too critical of the PTO over-reading, as the CAFC opinion in *Lowry* does not expressly focus on this difference with *Warmerdam*. See *Lowry*, 32 F.3d at 1583-84 (Fed. Cir. 1994) ("the claims require specific electronic structural elements" and "Lowry does not seek to patent the Attributive data model in the abstract.") (citing to *In re Bernhart*, 417 F.2d 1395 (C.C.P.A. 1969)). But see, *id.* at 1583 (noting the Lowry claim requires implementation by a computer).

259. See discussion *supra* notes 174-176 and accompanying text.

260. See discussion *supra* notes 174-176 and accompanying text. The shortcomings of the approach taken in the *Guidelines* discussed below apply equally to requiring computer hardware in machine claims.

within the fine arts, business, and the like, as well as the exclusion of laws of nature and abstract ideas as technologically inchoate.²⁶¹ If using software as language can serve as a vehicle for expressing these concepts and at the same time serve as the basis for obtaining § 101 approval, there exists a serious potential for preemption beyond the intended scope of the patent laws.

The Guidelines' reliance on the inclusion of computer readable medium is an unreliable talisman that the claims contain the necessary technology focus. The physicality of claims to software on a computer readable media is appealing. The combination has the look and feel of an article of manufacture. It can be inserted physically into and read by a general purpose digital computer. However, to rest the logic of the software as article of manufacture approach on the physicality or computer readability imparted by the carrier medium leads to improper results.

There can be no disagreement that merely putting software on a diskette does not magically transform the software expressed functionality into technological apparatus covered by § 101. This has no more effect than writing an idea in a book.²⁶² We would never confuse the patentability of a book, as a book, with the patentability of the ideas expressed if the language used were English.²⁶³ In the same way writing an idea into a book does not make the expressed ideas an article of manufacture, writing software onto computer readable media does not transform the combination of the media and the contained ideas into a unitary article of manufacture.

The Guidelines do not, however, predicate the § 101 determination solely on the physical writing of the software on the tangible medium. Addressing the PTO Board majority concern in *Beauregard* that software, as such, only has "the potential to control a machine,"²⁶⁴ the Guidelines state the presence of the computer readable medium "permit[s] the computer program's functionality to be realized."²⁶⁵ The Guidelines assert this makes the software claims statutory.

261. See discussion *supra* notes 208-227 and accompanying text.

262. This may, however, make the expression copyrightable. See *infra* note 285 and accompanying text.

263. In this situation, where the applicant seeks to cover the idea expressed in the guise of claiming the physical form (the book), the "printed matter" exception really is not that far off the mark. The issue is, in fact, separating the message from the medium. In making software patentability determinations the question is whether the particular claims to the message are patentable. As demonstrated in the text, the presence of the medium does not help answer that question. There is a good argument, however, that it would be better to discard the § 101 printed matter exception and address the issue under novelty rather than patentable subject matter. See *infra* note 274 and accompanying text.

264. See *supra* notes 252-253 and accompanying text.

265. See *supra* notes 174-176 and accompanying text.

The essential requirement of distinguishing between software used as language and software claimed as implementation, however, is to ensure we treat only the latter, technological application of the software functionality as § 101 apparatus. The former software as language claims must be separately assessed to determine whether what is being expressed is itself statutory or non-statutory subject matter. The requirement that the software be written onto media which can be put into and read by a computer does not draw this line. This requirement only means that the software functionality *can be* entered into the machine. It says nothing at all about whether it *must be*.²⁶⁶ The requirement, therefore, precludes neither coverage of other uses the software in the claims nor the resulting preemption of the making, use or sale of the expressed functionality beyond the claimed computer system implementation.

Two examples demonstrate the problem. First, as the examiner and PTO Board majority in *Beauregard* noted, broadly drafted computer readable media claims cover software written on a piece of paper.²⁶⁷ There is no difference in technical terms between such a claim and software written on a diskette. They are both machine readable; the former with the aid of an appropriate scanner and related software, the later using a disk drive and related software. In either case after a computer reads in the program instructions, to actually "execute" the program, it must be loaded into "active" memory.²⁶⁸ Thus, the diskette carried software is no more directly machine implementable than the paper carried version. The difficulty, of course, is that if a "paper inclusive" media claim contains no limitation to computer system implementation of the software functionality, it would be infringed by anyone making, using or selling a description of the software's functionality (its algorithm) written on paper. This would be true whether or not the paper version was intended for execution by a computer system.

Second, consider creation of a computer readable file describing the software functionality for the purpose of forwarding it electronically (by e-mail, for instance) to pass those ideas on to someone for their review and comment. Once again, a media claim to the software functionality would be infringed by the act of creation and use of this computer readable file even absent any actual use of the expressed functionality as software instruction in a computer system implementation.

266. Although the computer readable media formulation strongly implies the intended use is in connection with a computer, this is an insufficient basis for deciding the claims are to technology. See discussion *infra* notes 267-270 and accompanying text.

267. See *PTO Board*, *supra* note 156, at 6-7.

268. See L. Hollaar, *Justice Douglas was Right: The Need for Congressional Action on Software Patents*, 24 *AIPLA Q. J.* 283, 293-95 (1996).

Two counter arguments can be offered in an attempt to demonstrate that these examples do not pose real difficulties. First, the computer readable media formulation required by the Guidelines strongly implies the intended use of the software functionality is limited to computer systems. A reviewing court could simply invoke the "reverse equivalents" doctrine, draw its own inferential limitation to the claims, and hold that use of the software expressed functionality outside a computer application did not infringe. Second, because infringement only occurs by documenting the expressed functionality on "machine readable media," anyone wishing to use that functionality is free to do so provided they ensure it is never written down in tangible form.

Neither argument is persuasive. In the former situation a potential infringer must rely on the uncertainty of court interpretation: a slim reed to rely on in face of a potential infringement lawsuit. In the latter situation the usefulness of the undocumented approach in the real world is extremely questionable given the problems associated with the introduction, implementation, and particularly the on-going use of even fully documented procedures. Nor is it appropriate given the growing pervasiveness of electronic forms of communication. There seems little valid reason to force third parties to rely on potential court created exceptions or careful use restrictions to carve back grants of overly broad patent claims. Far better to draw an express limitation which properly focuses directly on what is to be accomplished: separating use of software as technology from use of software as language.

Unfortunately, the presence or absence of machine readable hardware permits claim coverage to go beyond the technological software as implementation to more general coverage of the underlying functionality expressed. The result is precisely what the Guidelines were trying to avoid. An idea or process, which may or may not be patentable subject matter, becomes classified as a statutory article of manufacture *per se* simply by using computer software language to describe it and putting that description on computer readable media.

The Guidelines, in fact, do recognize this possible "smuggling in" difficulty with the media format. The examiner is expressly instructed to examine claims which are not limited to specific hardware or hardware and software combinations (covering instead use on "any and every" hardware configuration) based on the underlying process (the functionality expressed).²⁶⁹ This desire to distinguish between process and apparatus claims is driven, not surprisingly, by the same motivation driving the need to distinguish software as language from software as implementation: ensuring that general claims to process (software as language expressing functionality) are not masquerading as apparatus (software as

269. See discussion *supra* notes 177-83 and accompanying text.

technological implementation). The distinction the Guidelines attempt to draw is appropriate. The difficulty, however, is that the "any and every" approach misses the mark because it is looking in the wrong place for the distinction. The determination does not rest on whether the claims cover implementation of the claimed software functionality on all (any and every) hardware platform or only on specific computer hardware as the Guidelines would posit it.²⁷⁰ It rests on whether the claims restrict the software functionality to implementation as part of a computer system, whether a specific system or generally. The result is that the "any and every" formulation overstates its case and may improperly preclude deserving software as implementation claims from receiving apparatus treatment under § 101.²⁷¹

4. *Syllogism Two (Under-inclusion)*

The Guidelines also require, as reflected in the second syllogism, that software claims include computer readable media to be classified as articles of manufacture. This position rests on the premise that because computer readable media is sufficient, or at least necessary, to permit the computer program's functionality to be realized (thereby demonstrating the useful arts nature of the claim), the absence of computer readable media should reliably identify claims which do not. The truth is otherwise. A brief look at the basics of the underlying computer technology demonstrates why.

In very basic computer science terms a special purpose computer system combines specific software instruction with appropriate hardware. Performing the combination and implementation of the software functionality²⁷² requires a number of steps. Taking a simplified case as

270. The CAFC cases holding that "means for" apparatus claims should be reviewed based on the underlying process are not based on the *total absence* of supporting structure not a finding that the claim preempts the process on all computer hardware. See *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1371 (Fed. Cir. 1998); *In re Alappat*, 33 F.3d 1526, 1540-41 (Fed. Cir. 1994) (en banc). The *Guidelines'* "any and every" hardware implementation analysis makes the same error as was made in *Benson*. Merely because a claim preempts use of the software on any and every computer hardware configuration does not mean it seeks to preempt a law of nature/abstract idea. *Gottschalk v. Benson*, 409 U.S. 63, 71-72 (1972). The Court immediately recognized the difficulty with the analysis in *Benson* and corrected it in *Flook* which criticized the approach. See *Parker v. Flook*, 437 U.S. 584, 590 n.11 (1978). See *infra* notes 299-300 and accompanying text. Note that this approach also runs counter to the direction of technological development with most software developers seeking to make their implementations as hardware independent as possible. See discussion *infra* note 277 and accompanying text.

271. Under the *Guidelines*, when the "any and every" test applies, the result is essentially to return to pre-*Alappat* process evaluation and the difficulties of that regime. See discussion *supra* notes 91-93 and accompanying text.

272. The discussion in the text focuses on implementation of software functionality. The basic software functionality itself is created when the software is created. Different

an example (there are many permutations), the following might occur. First, the software program must be loaded into appropriate hardware. This can be done by putting the software program on a computer readable medium. Second, the software is transferred from the medium by reading it (using appropriate hardware and software) into system storage, either a hard disk or directly into main memory. Finally, the program is "run" by moving the software instructions from storage into the operating registers (another form of computer memory) and executing them, all in accordance with the system's operating algorithm.

Viewed against this backdrop, the presence of computer readable media in the claims is neither sufficient nor necessary to computer system implementation of the software functionality. Putting the software on a diskette or other computer readable media can be a part of the overall implementation process. However, it is not sufficient as the entire process, not merely the entry step,²⁷³ is required to realize the software's functionality in the sense that the special purpose system actually performs the specific instructed task. Nor is there anything necessary about putting program code on a separate computer readable medium. The same computer instructions could be typed or read in orally one at a time or hardwired with the same effect.

The mere absence of computer readable media in a software claim tells us too little about whether the claim reaches beyond computer system implementation of the software's functionality. Syllogism two, therefore, draws a too restrictive, under-inclusive policy line. The result is a failure to apply the patent incentive to the full range of activities we wish to encourage.²⁷⁴

levels of "functionality" can be created at different times in the software development process. When the software design specification is created, it may indicate only the general approach. As a systems analyst works on the project the methodology may become more detailed. When a programmer drafts program code, more specific implementation details of the algorithm may be filled in. Provided the requirements of § 112 are satisfied (see discussion *supra* notes 65-67 and accompanying text), the particular level of detail at which the software invention is claimed (narrative, flowchart, program code) is irrelevant to the patentability inquiry. What matters for purposes of that inquiry is that the software claim requires implementation of the software functionality as a component of a computer system.

273. Unless the claims are directed specifically to the method of computer access to the program code itself, in which case the manner of entry is the invention. This specific situation is very different from a claim to the software expressed functionality itself. For purposes of determining the patentability of the software's expressed functionality, it does not matter how the software is entered, whether by diskette, ROM, download from a network or typed from a keyboard or read in orally a line at a time.

274. See, e.g., *In re Trovato*, 42 F.3d 1376, 1383 (Fed. Cir. 1994), *vacated and remanded*, 60 F.3d 807 (Fed. Cir. 1995) (en banc) (applicant arguing "scientists in the field of computer science are not less worthy of obtaining patent protection for their inventions than technologists in more traditional fields."). It is possible non-media software claims which fail to

The Guideline computer readable media requirement, therefore, misses the patentable subject matter target on all counts. As discussed in connection with syllogism one, its presence does nothing to ensure the claimed functionality is covered only in its technological role as instruction for a computer system implementation. As just discussed, its absence does not mean the claims go beyond such coverage.

Because the requirement fails to achieve any useful purpose, it only adds irrelevancies to the software claiming process. The most obvious technical effect is to create a groundless rejection under § 101. One could argue this is of no practical significance. By proper drafting, the computer readable media requirement can easily be met.

However, forcing software claims into this unjustified format has two less obvious but far more serious ramifications. First, the technological advance in software implementation inventions is in the functionality imparted to the related special purpose computer system. Protecting this functionality solely when carried on a computer readable medium unduly limits the scope of the protection provided and, consequently, the incentive afforded to these inventive efforts. In particular, the requirement limits infringement only to situations when the competitor makes, uses or sells the software in combination with a machine readable medium.²⁷⁵ This has undesirable consequences as variations occur in the

receive article of manufacture status will survive the process tests contained in the *Guidelines*. See *supra* note 179 and accompanying text. There is, however, no reason to subject these claims to such review and risk inappropriate exclusion.

Additionally, although including computer readable media may be one way to indicate an appropriate claims limitation, there are other, broader, more reliable ways to achieve the desired result without improper exclusion of claims to software "as such." See *infra* Part III.G (applying the implementation versus language test).

It could also be argued that the inclusion of computer readable media in a software claim creates, by its mere presence, a qualifying § 101 article of manufacture (in effect doing away with the "printed matter" exception entirely and accepting the combined medium/software as a unit). The analysis would move immediately to questions of novelty and nonobviousness under §§ 102 and 103. This article argues that only novelty in the useful arts is sufficient to sustain an application under § 102. Consequently, a point of novelty inquiry must be made to determine if appropriate novelty is present. See discussion *infra* notes 331-32 and accompanying text; see Moy, *supra* note 242 and accompanying text (proposing a similar analysis when dealing with such hybrid inventions). As the only novelty would lie in the software in such media-article of manufacture claims, a useful arts inquiry into the nature of the claims to the carried software would, therefore, still need to be performed. This would require resolution of the same issues faced regarding whether software claims are patentable subject matter under § 101. Although the result would be the same if the proposed implementation-language test were applied under § 102 as part of this inquiry, it is better to ignore the media entirely and make the determination directly.

275. Granted this is an improvement over prior practice which primarily dealt with software in process terms, meaning frequently direct infringement actions could only be brought against end-user customers, with competitors only reachable under the more difficult contributory infringement claims. Part of the impetus for the article of manufacture

way software is recorded and delivered. Even if an expansive reading of computer readable media is accorded the claims,²⁷⁶ the need to determine whether the competitor's medium is included provides an uncertainty to enforcement, uncertainty breeds expense and expense reduces incentive.

Although this may seem of only minor concern given current technology primarily deals with software on relatively standard forms of computer readable media, there are already indications that this will not continue to be the case. Consider the situation in which the only copy a customer receives is down-loaded electronically from an Internet server located outside the United States. Does the provider infringe a United States media limited patent claim or is the patentee's only recourse against the consumer who creates and uses the new computer readable media version on receipt?²⁷⁷ Or, as human speech interfaces become more prevalent, does a media limited claim protect the functionality of software programs entered orally by a user? In both situations given the provider/distributor's activity is likely to be significantly more damaging to the operation of the patent incentive, and providing a direct remedy is certainly more efficient as well as commercially attractive, it seems inappropriate to restrict enforcement alternatives by what is an otherwise irrelevant hardware carrier limitation.

Second, forcing software claims into a "hardware" based mold in order to clear § 101 operates against the evolution of the technology itself. Computer technology continues to move away from hardware implementations. A large portion of all new development is done exclusively in software. As improvements are made, new versions or replacement products substitute software implementations for existing hardware (including hard-wired) solutions. Additionally, the software solutions

claim format was to provide this additional protection to software related inventions by allowing the patent holder to attack infringements involving producers or distributors of media loaded with the program. See discussion *supra* note 92 and accompanying text. Ironically, as discussed below in the text, even the additional coverage provided by the current article of manufacture "hardware" claim format may eventually prove inadequate in the face of further technological advances eliminating the need for carrier based distributions.

276. Like that posited by the examiner and the PTO Board majority in *Beauregard*. See *supra* note 66 and accompanying text (discussing means for claims); see discussion *supra* note 267 and accompanying text.

277. The software at either end will "come to rest" in a computer readable medium and could be attacked. This does not help the patentee obtain relief against the producer/distributor unless either she holds a local equivalent patent or, more problematically, the "over the line" transmission is deemed either a "copy" in its own right (a "carrier wave" version, see discussion *infra* note 289 and accompanying text) or an infringing import into the United States under 35 U.S.C. § 271(a) (1994). The situation becomes even more complex if (as) technological advance eventually permits systems to create new software or copies on demand without the need for a permanent computer readable copy.

themselves are becoming increasingly independent of their hardware operating platforms. Many software implementations are designed expressly to be transportable and inter-operable with as large a range of hardware alternatives as possible. Basing software as article of manufacture approach on the presence of hardware (even computer readable medium), therefore, forces computer technology innovators to switch paradigms when it comes to obtaining patent protection for the results of their inventive efforts. The effect is to retard the advance of computer technology rather than to encourage it.

G. A NEW TEST AND A NEW SYLLOGISM

The foregoing analysis demonstrates there are serious shortcomings with where the Guidelines take us²⁷⁸ with regard to software as article of manufacture claims. The presence of machine readable media is neither a necessary nor a sufficient condition to article of manufacture apparatus treatment of software inventions.²⁷⁹ In fact, the hardware re-

278. The U.S. is not alone in this enterprise, or in having conceptual difficulties with how to deal with software related inventions. The European Union is currently re-examining the issue of patentability of software under the European Patent Convention ("EPC") of Oct. 5, 1973, TS No. 20 (1978), 13 I.L.M. 270. Currently the EPC expressly states that computer programs "as such" are unpatentable. EPC art. 52(2). Court interpretations of what constitutes a computer program as such, have pointed in both directions. *Compare VICOM/Computer-related inventions* (Decision T 208/84)[1987] (Technical PTO Board of Appeals found the program must involve a technical contribution to a technical field, such as a substantial increase in processing speed, computer software driven image enhancement patentable) with *Merrill Lynch's Application* (1989) RPC 561 (U.K. opinion that claims to a computer with software to automate making a market in securities not patentable). The Mar., 1997, U.K. Court of Appeals decision in *Fujitsu* is interesting in that the court appears to reject entirely the concept of software as implementation under the "as such" rationale. *In re Fujitsu*, Application No. 9204959.2 (C.A. 1997) Finding against the applicant, which was seeking to patent a chemical computer aided design software system, the court reasoned:

[Counsel for applicant] is right that a computer set up according to the teaching in the patent application provides a new "tool" . . . which avoids labor and error. But those are just the sort of advantages that are obtained by the use of a computer program. The fact that the patent application provides a new tool does not solve the question of whether the application consists of a program for a computer as such or whether it is a program for a computer with a technical contribution. . . . I believe the application is for a computer program as such The only advance is the computer program which enables the combined structure to be portrayed quicker.

Id.

The Court, operating under the EPC's "is capable of industrial application" test, rather than the U.S. Constitution's "useful arts" requirement, is obviously unimpressed by the technical contribution made by the new tool. *Id.* Apparently the industrial application standard under the EPC (or at least in the U.K.) is higher than the technological arts standard in the United States.

279. Including computer readable media in the claims may be used by the applicant as a vehicle to create the necessary and sufficient restriction of limiting the claimed software to

quirement in the Guidelines adds nothing to the patentable subject matter analysis but irrelevancies and distractions. It should be abandoned and a new paradigm adopted.

This same discussion also indicates the overall patentable subject matter inquiry should be refocused. The question is not "if software is patentable," but whether particular uses of software should be considered patentable subject matter in light of the incentive to the useful arts policy objectives of the patent laws. The general answer to this question is straight-forward: when software functions as technology it should be, when it does not, it should not. To draw this line, the patentable subject matter test must separate claims to software expressed functionality as instructions to a computer system²⁸⁰ from claims which use software as a language to express the described functionality.²⁸¹ Only the software implementation claims fall directly within the useful arts.²⁸² The software as language claims may or may not involve patentable subject matter depending on the nature of the particular underlying, software independent, functionality expressed. What is required is a test which reliably differentiates between these two roles of software in specific claim contexts.

Such a test can be simply drawn. Placing the dividing line between claims which limit coverage of software functionality to use in a computer implementation and those that do not provides a workable basis for making the "useful arts" patentable subject matter determination.²⁸³ When claims specifically limit coverage of the software expressed functionality to use as the instructional component of a special purpose computer system they only cover software expressed functionality in its

use a computer system component. Whether this has been accomplished is a matter of claim interpretation. See discussion *infra* note 340 and accompanying text. This use is not, however, the same as the *Guideline's* mere presence test, which does not, standing alone, ensure an appropriate limitation.

280. Software implementation should not be confused with the display of software program code by a computer system. The display of the program is very different than the execution of the software instructions themselves. Even though in both cases a computer is involved, the display use treats the software merely as unpatentable data. See *supra* note 242 and accompanying text.

281. The fact that a person may not be able to directly read the particular code is not a reliable test. There are a number of ways for people to access software functioning as language, not the least of which is by reading the patent itself. See discussion *supra* notes 244-46 and accompanying text.

282. Cf. *In re Toma*, 575 F.2d 872, 877 (C.C.P.A. 1978) ("method of enabling a computer to translate natural languages is in the technological arts, i.e., it is a method of operating a machine").

283. Cf. *Hollaar*, *supra* note 268, at 279 (approaching the issue from a similar direction but viewing software "as such" claims as to the process but advocating amending the Patent Act to make such processes *per se* statutory subject matter).

technological implementation role.²⁸⁴ If the claims are not limited to the computer implementation role they move beyond software as technology into the realm of software as communication device. In these situations it is not the method of expression which carries the useful arts inquiry, but the nature of the functionality being expressed.²⁸⁵

284. Cf. *Toma*, 575 F.2d at 878 (“*In re* Benson . . . involved data processing methods useful in a computer, but not expressly limited to use in a computer.”).

285. There is clearly a connection between patent and copyright protection of computer software. It is important to keep each regime in its own territory. The patent law question addressed in this article is determining when the software expressed functionality is patentable subject matter. Whether or not that expressed functionality is patentable, as an article of manufacture or otherwise, software expression, as any other form of expression, also may be subject to copyright protection. See *PTO Board*, *supra* note 156, at 12 (citing *In re* Yardley, 493 F.2d 1389 (C.C.P.A. 1974)); *Apple Computer v. Franklin*, 714 F.2d 1240, 1247 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984); 17 U.S.C. § 102(a)(1) (1994) (covering literary works under copyright and interpreted by H.R. Rep. No. 1476, 94th Cong. (1976) to include computer programs).

Faced with the problematic “mathematical algorithm” and FWA test for software patentability, the computer software industry has been testing the border between the realm of copyright and patent in recent years. See *Lotus v. Borland*, 49 F.3d 807 (1st Cir. 1995), *aff’d by an equally divided court*, 516 U.S. 233 (1996); *Computer Assocs. v. Altai*, 982 F.2d 693 (2d Cir. 1992); *Whelan Assocs. v. Jaslow Dental Lab.*, 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987). To maintain the integrity of the two systems, however, the copyright protection must stop at the specific expression rather than implicating the underlying functionality. This need is reflected the (in)famous “idea versus expression” dichotomy which lies at the heart of copyright law. See *Baker v. Seldon*, 101 U.S. 99, 104-05 (1879) (“The object of one [expression] is explanation; the object of the other [idea] is use”). This idea versus expression dichotomy should not be confused with the language versus implementation dichotomy proposed in this article for patents. The patent distinction seeks to determine whether claims to the *functionality* contained in the software should be assessed for patentable subject matter purposes as claims to a process (language) or to apparatus (implementation). The copyright test looks instead to whether the *form of protection* should be patent or copyright, protecting the critical difference between copyright protection for the specific program code expression and patent protection for the underlying functionality. *Id.* (“The former [expression] may be secured by copyright. The latter [ideas] can only be secured, if it can be secured at all, by letters patent”).

In both *Computer Assocs.* and *Lotus* the reviewing Courts of Appeal appear to have avoided the pitfalls of *Whelan* and properly rejected copyright coverage of functionality (methods) and directed them toward patent law by applying the idea-expression test. Of particular interest is the reviewing courts’ expansive view of the copyright “merger” doctrine to capture concepts like efficiency in coding as uncopyrightable but potentially patentable functionality. See *Computer Assocs.*, 982 F.2d at 703-05 (citing *Baker* and the underlying policy considerations for limiting copyright to expression) and *Lotus*, 49 F.3d at 815-17 (citing *Baker* and noting the 17 U.S.C. § 102(b) (1994) exclusion of “methods of operation” from copyright law).

Speedy resolution of the continuing confusion on the patent side of the equation can do much to help the courts continue to keep copyright in its proper realm. A failure to do so risks distorting not only patent law but copyright law as well. See Dennis S. Karjala, *The Relative Roles of Patent and Copyright in the Protection of Computer Programs*, 17 J. MARSHALL J. COMPUTER & INFO. L. 41 (1998).

That this dividing line holds up for normative purposes can be demonstrated by analyzing the results achieved by separating these two types of software claims. First, straight-forward analytic progression shows that claims covering software only as computer system implementation are limited to matter falling within the technological arts. Claims covering general purpose computer hardware are clearly machines within the useful arts and the § 101 patentable subject matter categories. Claiming a computer system consisting of a combination of such a general purpose computer hardware plus specific software is tantamount to claiming a special purpose computer. There simply is no logical reason for finding that the inclusion of software instruction in an otherwise patentable general purpose computer system to permit performance of a specific task makes the combination special purpose computer non-patentable subject matter.²⁸⁶ Such a special purpose computer is also a machine within the technological arts and § 101.²⁸⁷

Once one accepts the special purpose computer system analysis, it is a short and unavoidable step to finding that claims to the software itself, when limited to its "instruction" role as a part of the special purpose computer system, constitute both technology and apparatus (machine or articles of manufacture). The patentable special purpose computer system can be viewed as comprising two separate components: the computer hardware and the software as the implementation "instructions" for op-

I am indebted to Professor Karjala for helping me to focus on this important relationship between patent and copyright law.

Clarifying the proper domains for copyright and patent protection of software inventions may also obviate the need for *sui generis* protection for these inventions. Providing copyright protection of the "expression" (insulating the effort of doing the coding from easy misappropriation) and appropriate patent protection for the functionality (protecting truly novel advances in structure or operation) would seem to pretty much cover the field. The problematic area of graphical user interfaces (the source of the contest in *Lotus*) provides a good example: The functionality (operation) of the interface will be eligible only for patent coverage subject to the rigorous patent requirements of novelty and non-obviousness. Subject to merger, the original coding and any non-useful icons or graphics, will be protected in the more limited form of copyright, which prohibits copying but permits independent creation.

This analysis assumes, of course, that we have determined that the patent incentive is appropriate to software at all. See *supra* note 238 and accompanying text. Depending on the outcome of that more general discussion, it may well be that some form of expedited application process with more limited review for novelty and nonobviousness, a shortened term and/or limitation on the exclusive right granted (for example, not precluding independent creation) may be appropriate adjustments resulting in a *sui generis* modified patent or copyright regime for software. The burden remains on industry, however, to show why software is, in fact, different. *Id.*

286. Certainly if the same machine were hardwired with the same "software" instruction, there would be no argument it was not patentable subject matter.

287. See discussion *supra* notes 73, 104-33, 229 and accompanying text.

erating that hardware.²⁸⁸ This is entirely consistent with our view of other machines consisting of various components. Just as we unhesitatingly treat each component of other types of machines as a separately patentable article of manufacture, there is no reason not to do so in the case of computer systems. Clearly each of the general purpose hardware "components" of such a system can stand alone as patentable subject matter. And when the software component of the system is claimed only in that component role (by expressly restricting coverage in the claims to use as computer system instruction), it too should be treated as separately patentable subject matter.²⁸⁹

In contrast, claims which do not clearly limit coverage of software expressed functionality to its implementation function merely employ a software context as the vehicle for expression of the underlying functionality. The particular choice of software as the vehicle to convey that information to others adds nothing more to the statutory subject matter analysis than would expressing the same ideas in English, Italian or Japanese. Nor does stating in the detailed description that the functionality being expressed in the claims *may*, even in the preferred embodiment, be written in software code and used in a computer, exclude coverage of other non-computer system uses of that functionality by the claim. In these situations it is the expressed functionality itself which must be assessed for patentability, not the particular form of its expression.²⁹⁰

288. Cf. McCutchen, *supra* note 88, § 4 (pointing out that it is short jump from disk plus data structure to disk plus program).

289. This is essentially the "software" as cam (component) approach raised by IBM in *Beauregard*. PTO Board, *supra* note 156, at 14. The "software as implementation" formulation proposed in this article takes the analogy one step further with the software functionality itself serving as the cam, with no need for the irrelevant and confusing "carrying" diskette. See discussion *supra* notes 272-77 and accompanying text; see *infra* notes 312-14 and accompanying text. Permitting claims to software "as such" also eliminates the concerns over innocent telecommunications and network service provider liability under the carrier wave "signal" variation of the current article of manufacture approach. See Stern, *supra* note 93. Under the signal variant, the provider may make technically infringing copies of the software in signal form as part of the transmission process even without knowledge of what the signal contains. Eliminating the carrier signal format in favor of a direct claim to the software functionality when used in its implementation role as part of a computer system takes copies created solely for mechanical transmission purposes outside the patent's reach. A telecommunications provider could, of course, become liable for contributory infringement if they act with intent to assist an infringing sender or receiver. See *supra* note 92 and accompanying text.

290. This dividing line eliminates confusing the fact that a claim to an idea expressed in software language involves the useful arts merely because of the choice of the software context. See P. Weissman, *Computer Software as Patentable Subject Matter: Contrasting United States, Japanese and European Laws*, 23 AIPLA Q.J. 525, 548 (1995). In the software as language cases, whether the ideas are patentable subject matter turns exclu-

The above approach can be captured by replacing the Guideline's hardware focus with a new software "as such" article of manufacture test based on claims which only cover the software functionality when used as a computer system component. This suggests the following new syllogism: machine components are patentable subject matter (articles of manufacture). Claims which cover software functionality only when used as a component of a computer system implementing the task instructed by the software, are claims to machine components. Therefore, such software claims are claims to articles of manufacture and are patentable subject matter as such.

This test does not preclude an applicant from making broader claims couched in software language terms. The objective of the test is only to ensure that it is clear how the software is actually being used in a particular claim. Specifically, the test seeks to avoid confusing broad claims to the underlying ideas expressed in software language terms with software as technology claims resulting in inappropriately superficial review of those broader claims under § 101. If an inventor believes she is entitled to coverage of the expressed functionality beyond the software implementation of her invention in a computer system, she should seek it. But in those instances, she will not be permitted to rely on the software expression of the functionality for purposes of the § 101 inquiry.

1. *Does it Work? Testing the New Syllogism for Consistency With Patent Law Objectives and Practicality*

Having proposed a substantially "bright line" test for determining if particular software claims should be treated as patentable subject matter, it is necessary to evaluate its efficacy. This requires the test work on three levels. First and foremost, its application must only identify as patentable subject matter inventions which are consistent with the objective of the patent laws: incentivizing exclusively those activities which fall within the technological arts. Second, it should not impair, and if possible should enhance, application of the additional requirements for patentability, in particular the "progress" objectives embodied in the requirements of novelty and nonobviousness. Third, it must work in practical terms, permitting efficient and effective drafting and examination of patent applications and appropriately flexible enforcement options.²⁹¹

sively on the nature of the idea not the method of its expression. The analysis should simply ignore the use of program code or software context as the method of expression.

291. It must also be integrated with other aspects of the patent laws such as notice requirements, first sale and the like. See discussion *infra* note 341 and accompanying text.

a. Patentable Subject Matter

The first criterion requires the test clearly identify and separate software claims which involve subject matter falling within the technological arts from those that do not. Put another way, it should draw a dividing line which avoids either over-inclusion or under-inclusion. To avoid over-inclusion, the test must prevent software patents from reaching outside the useful arts to cover innovations in the fine arts, in business and in the social and physical sciences (including claims to newly discovered laws of nature and natural phenomena) or abstract ideas (claims to concepts without adequate technological implementation).²⁹²

The key is to realize that the problem of software claims reaching outside the technological arts arises in connection with claims to the non-patentable subject matter "as such," as opposed to claiming computer technology tools useful in those fields. Merely because a computer system is useful in automating tasks in a non-technological activity (be it translating, accounting, selling or analyzing social or physical science data) does not make the machine itself any less a part of the technological arts.²⁹³ The tool is technology and the resulting increases in speed, accuracy, resource efficiency and the like arising from its use are technological advances. Therefore, the objective is to separate claims to computer technology (apparatus) from claims which reflect innovations in the underlying field of application.

This is precisely the issue addressed and resolved by the express dictates of the proposed test. By requiring that the claim to software functionality be limited to its role as part of a computer system to obtain patentable subject matter status as such, the test eliminates the software as language claims which pose the over-inclusion difficulty.²⁹⁴ When the limitation is present the expressed functionality is preempted by the claims only to the extent of its use in implementing the related computer system.²⁹⁵ There is no risk or confusion that the functionality

292. See discussion *supra* notes 45, 212-26 and accompanying text.

293. See, e.g., *In re Freeman*, 573 F.2d 1237, 1247 (C.C.P.A. 1978) ("A claim drawn to new, useful and unobvious apparatus, specifying what that apparatus is, and not merely what it does, would not, other the other hand, be rejectable on the sole ground that the only presently known use for that apparatus is the practice of an unpatentable method"); *In re Johnston*, 502 F.2d 765, 771-76 (C.C.P.A. 1974), *rev'd on other grounds, sub nom. Dann v. Johnston*, 425 U.S. 219 (1976) ("Record-keeping machine systems are clearly within the 'technological arts.'"); *In re Toma*, 575 F.2d 872, 877-78 (C.C.P.A. 1978) (Translation is not a useful art, but a computer implementation is).

294. It has the added advantage of doing so without the problematic labels of "doing business," "mental steps" and the like which, despite criticism, refuse to disappear. See discussion *supra* notes 208-10 and accompanying text. Justice Frankfurter would approve. See *supra* note 229 and accompanying text.

295. This is no different than any other machine which preempts its included operating algorithm. As the courts have noted, the fact that all machines work "algorithmically" (per-

is covered in broad terms as such. If the limitation is absent then software is being used to describe and obtain coverage for the underlying functionality in general application. In these cases the expressed functionality should be assessed directly and independently of the software expression under § 101 in the same fashion as any other non-software claim.

Two examples demonstrate. First, consider software styled claims to a system for pro rated allocation of mutual fund assets and costs similar to those at issue in the *State Street Bank* case.²⁹⁶ The invention clearly concerns an activity connected with the operation of a mutual fund business, a non-technological art. In order to pass the proposed implementation test, the claims must only cover use of the methodology as software instruction for a related computer system. Provided they do so, such claims do not prevent others from using the underlying allocation methodology in any other context. Nor do such claims extend to the underlying operation of the mutual fund business or even to the cost allocation activity "as such." They only cover the particular computer tools for implementing the allocation methodology. In this form, the software claims should be considered both useful arts and a statutory article of manufacture.

If, however, the limitation is absent then the claims reach beyond the computer implementation of the software expressed methodology, pre-empting its use by others more generally. In this case any potential software implementation should be ignored and the underlying asset-cost allocation methodology itself assessed to determine if it falls within the useful arts and constitutes patentable subject matter.²⁹⁷

haps more noticeably in the case of computer software) does not make them non-statutory subject matter. See, e.g., *Diamond v. Diehr*, 450 U.S. 175, 189 n.12 (1981) (if carried to the extreme no invention is patentable as all inventions ultimately rest on laws of nature). See discussion *infra* note 300 and accompanying text.

296. *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368 (Fed. Cir. 1998). See discussion *supra* notes 108-23 and accompanying text.

297. The waffled conclusion stated in the text reflects the much broader question of how far we should go in treating process claims as patentable subject matter. In *State Street Bank*, the CAFC applying the "useful, concrete and tangible" output test seems to indicate the asset-cost allocation process might be. *State Street Bank*, 149 F.3d. at 1373. See discussion *supra* notes 108-23, 229 and accompanying text. Under the implementation-language test this issue is not relevant to software as patentable subject matter inquiries. Software as implementation (apparatus) is patentable subject matter. Claims using software as language must stand or fall on the underlying functionality and do not involve software patentable subject matter questions, avoiding the software context clouding that issue being precisely the point of the test. Although the broader process issue is beyond the scope of this inquiry, it is worth noting that applying the CAFC's "useful, concrete and tangible" output test to those and other process claims will have far reaching consequences which should be explored. It apparently validates the mutual fund allocation process involved in *State Street Bank*. The court's elimination of the method of doing business doc-

Second, consider claims to software implementation of the formula "F=ma." The formula, of course, represents a law of nature²⁹⁸ and under current Constitutional policy should not be patentable subject matter even in favor of the discoverer. If the claims only cover computer software functionality when serving as a component of a computer system there is no general preemption of the F=ma formula beyond its operation as the software component of the computerized acceleration calculator. The world remains free to use the formula's functionality in any other application. In this form the claimed software functionality is technology and patentable subject matter.²⁹⁹ In the absence of the limitation, however, then the formula itself must be analyzed without consideration of any software form of expression. Clearly, the law of nature does not involve patentable subject matter.

These examples raise two concerns. First, if the allocation method (or, less likely, the acceleration calculator) can only practically be implemented using a computer system, granting the patent appears to *de facto* preempt the incorporated and unpatentable functionality. Even though others may be free in theory to use the underlying methodology, in prac-

trine makes it appear that the test might also validate any business method which involved creating "useful, concrete and tangible" outputs. *Id.* at 1375. There appears to be no sound basis for stopping there; the logic justifies the validation of any process provided the process output meets the "useful, concrete and tangible" requirement. Perhaps the court's focus on transformations "by a machine" is intended to mean only processes which use apparatus satisfy the test. *Id.* at 1373. The question is whether the analytic framework matches the useful arts limitation of the Constitution. To make that determination requires closer scrutiny of the requirements. That issue must be left for another day.

298. Note that F=ma in the Newtonian sense is different than the equivalent computer software code statement "f=ma." The former expresses a specific law of nature where the variables stand for particular natural phenomena (force, mass and acceleration), the latter expresses a pure mathematical equality between variables which are undefined ("f" is the product of the contents of variable "m" and "a" regardless of what they stand for). Without more it is impossible to tell if a claim to the equation expressed in software terminology/context preempts the natural relationship, the general mathematical equivalence or merely a specific computer implementation of one or the other. The ramifications are very different. See *infra* note 299 and accompanying text. This ambiguity is in large part why the mathematical algorithm test proposed in *Benson* and still alive in the FWA refinement causes so much trouble in the software context. See *supra* note 85 and accompanying text. The more definitely preemptive Newtonian version is at issue in the example.

299. See discussion *supra* note 293 and accompanying text. Cf. *In re Bradley*, 600 F.2d 807, 812-13 (C.C.P.A. 1979), *aff'd by equally divided court, sub nom.* *Diamond v. Bradley*, 450 U.S. 381 (1981) (Burger, C.J., not participating) (claiming a machine is not the same as claiming any particular calculation it may perform; the former is statutory). Whether the acceleration rate output would satisfy the *State Street Bank* "useful, concrete and tangible" test is uncertain although it appears likely that it would. See *supra* notes 108-123, 229 and accompanying text. If it does not, we are left with the anomalous result that a machine does not satisfy § 101. If it does, the issue is how to draw the useful arts line in a way which prohibits claims to the underlying calculation process itself (the law of nature) being treated as patentable subject matter. See *supra* note 297 and accompanying text.

tice they simply cannot. Second, in both cases there appears to be a dramatic imbalance between the simplicity of the software implementation and the scope of the patent preemption. All the work seems to be in the underlying functionality, not in the computer implementation.

Neither concern is properly a patentable subject matter issue. Regarding the first, unquestionably the claimed invention "as a whole" is technology and under current policy should be included within the basic reach of the patent incentive. The fact that the only currently available technology for implementation is covered by the patent does not mean others are precluded from using the underlying functionality. The patent only prevents use of the specific implementation claimed by the applicant; others remain free, and encouraged by the patent incentive, to develop alternative technological means for accomplishing the task. This quest for additional useful art advance, and reward for achieving it, is precisely the point of the patent laws.³⁰⁰ The patent system is, therefore, working exactly as designed. The remedy, if one is required, lies in seeking a narrowing of the patent incentive system itself.

All that is required to deal with the second concern is to note that finding the claims involve patentable subject matter does not end the

300. Even if the claims cover the only practical form of implementation currently available, this does not mean issuing a patent preempts the underlying functionality. This was the concern expressed by the Court in *Benson*. See discussion *supra* notes 45, 76-80 and accompanying text. Merely because the claimed apparatus is currently the only method for implementing the underlying activity and, therefore, provides a *de facto* monopoly over that activity, does not mean others are *de jure* precluded from developing new apparatus. This *de facto* preemption is precisely what the patent law incentive is about. The inventor of the first solution has exclusive rights to prevent other from using that solution. Others are encouraged to dedicate themselves to discovering new technological implementations. Cf. *In re Johnston*, 502 F.2d 765, 771 (1976), *rev'd on other grounds, sub nom.* *Dann v. Johnston*, 425 U.S. 219 (1976).

The appealed apparatus claims are not drawn to cover either a method of doing business or even a method of bookkeeping . . . Obviously, banks would be free to expand their services and use any apparatus they may desire *except* the apparatus set forth in appellant's claims. We do not find that situation to be in conflict with [the Patent Clause], since '[t]he right to exclude others' is the very heart of our patent system.

Id.

In all cases, including *de facto* preemption situations, there must be technological novelty to support the issuance of the apparatus patent. Otherwise, claimants could use existing technology, such as computer automation, to obtain unmerited patents controlling advances in an underlying non-technological activity. See discussion *infra* notes 331-35 and accompanying text. This is the complement of the "function of a machine" analysis which denied process claims if there was currently only one apparatus identified for performing it. The CAFC eventually recognized that if the invention also includes discovery of a *useful arts* process, thus providing proper novelty support for preemption of the process as well, a patent should issue on that process as well as the specific apparatus for performing it. See *In re Tarczy-Hornoch*, 397 F.2d 856 (C.C.P.A. 1968); CHISUM, *supra* note 18, § 1.03[7] (discussing the function of a machine doctrine).

patentability inquiry. By passing through the § 101 gate the invention is only ensured a full review concerning the other elements of patentability. Closer scrutiny reveals that the proper basis for this concern is not that the software implementation of an accounting system or acceleration calculator instruction falls outside the useful arts, but whether there is sufficient inventive technological behavior to justify issuance of a patent based on novelty and nonobviousness considerations.³⁰¹ This, of course, substantially mitigates the first concern in both examples as well. Only true technological advances will be rewarded with a patent.

Over-inclusion possibilities related to abstract ideas present a slightly more complex analysis and deserve specific attention. In this arena the general policy objection is that the claimed invention has not been sufficiently "applied" to merit reward as progress in the technological arts.³⁰² On close inspection the "abstract idea" concern implicates two different kinds of abstraction. The first involves true § 101 "abstract ideas" concerns, those claims which completely fail to put the covered idea, good as it may be, to any specified technological use. The second kind of abstraction arises from excessive vagueness concerning how (versus in what context) the idea should be implemented. In these latter cases, which technically deal with § 112 enablement, although one or more technological applications of the idea may be clearly identified, there is insufficient technical detail to actually make it work.³⁰³

The policies behind both exclusions are related. In both cases the concern is giving a patent "too soon." The specific concerns are different, however. Granting a patent to inventions which are abstract in the § 101 sense risks impeding development of the entire range of specific technological applications of the idea.³⁰⁴ A patent covering an abstract idea in the § 112 sense impedes development of the specific technology required to actually effect the identified application.

Although related, these two sides of the abstraction coin have very different ramifications and should not be confused.³⁰⁵ The cases involv-

301. Cf. *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1375 (Fed. Cir. 1998). For many of these problematic cases, including the cases noted in the text, the novelty and nonobviousness requirements will pose a very substantial barrier to patentability. See discussion *infra* notes 322-25 and accompanying text.

302. See discussion *supra* notes 223-25 and accompanying text.

303. See *supra* note 65 and accompanying text.

304. If one chooses to view mathematical relationships as distinct from laws of nature (see *supra* note 276 and accompanying text), these types of claims can also be covered under the abstract idea heading. This is what the CAFC did in *State Street Bank and Alappat*. See *State Street Bank*, 1998 WL 409704 at *4-5 (Fed. Cir. 1998); *In re Alappat*, 33 F.3d 1526, 1542-43 (Fed. Cir. 1994).

305. See *infra* note 311 and accompanying text (providing an example of the overall abstract idea analysis). Note that an abstract idea may be fully enabled as in general claims to a mathematical model which is fully detailed and easily implementable in

ing the first abstraction concept, the lack of a specified useful arts application, are successfully dealt with by the test in the same fashion as other non-patentable subject matter situations. The test readily separates software claims raising this abstraction difficulty from those that do not. If the claims cover the software only when used as part of a computer system implementation there is by definition a specific useful arts application: use as a component of a computer system.³⁰⁶ Consequently, there is no patentable subject matter abstract idea concern. If, however, the claims do not contain the limitation then the software is serving in its language role. In these cases the software context is not relevant and the claims must be tested directly for lack of a specified useful arts application in the same fashion as any other potentially unapplied concept.³⁰⁷

Software as implementation claims, like all claims, must still be assessed under the second enablement abstraction concept.³⁰⁸ They are tested under § 112 to determine whether the specification is sufficient to permit a person of ordinary skill in the applicable art to create the appropriate software component and to use it to instruct the appropriate computer system to perform the claimed computerized task.³⁰⁹ Unless this

software by someone skilled in the art but with no indication concerning field of use (including in a software implementation). This is very different than a claim to a software component of a specific computerized marketing system based on the model but lacking adequate enabling details to permit a skilled computer scientist to actually implement the system. This latter issue was raised by Justice Frankfurter's concurrence in *Funk Bros. Seed Co. v. Kalo Inoculant Co.*, 333 U.S. 127, 134 (1948). See *supra* notes 224, 229 and accompanying text.

306. See, e.g., *In re Lowry*, 32 F.3d 1579, 1583 (Fed. Cir. 1994) ("Lowry does not seek to patent the Attributive data model in the abstract."); *In re Toma*, 575 F.2d 872, 877 (C.C.P.A. 1978). Note that for patentable subject matter purposes it is irrelevant that the specific field of application for the computer implementation be specified. The system itself is part of the useful arts, whether or not it has any useful application or whether applied to solve problems which are themselves in the useful arts. See discussion *supra* note 229, 293 and accompanying text.

307. The test makes it clear that in software as language cases, the software neither creates an abstract idea concern nor does it lend any technological underpinning thereby automatically creating a useful arts application of that functionality. See *supra* notes 229, 293 and accompanying text.

308. In terms of result, it does not matter if the § 101 or § 112 issues are addressed first. A reasonable argument can be made, however, that until at least the § 112 definiteness requirement has been satisfied is there no clearly identified invention to be tested under § 101.

309. Under the proposed test the *Guidelines* requirement that § 112 be satisfied by the presence of specific hardware disappears. See discussion *supra* notes 184-90 and accompanying text. Consistent with the *Guidelines* there is no requirement that the specification list specific software code. All that is required is that the software functionality as described in the application can be implemented by one skilled in the art as part of a computer system without undue experimentation. The more reliance on "one skilled in the art," the more likely the application will face a novelty or obviousness rejection. See discussion *infra* notes 322-25 and accompanying text. In addition, claims will only extend beyond

is possible the invention is not sufficiently mature to permit it to act as a machine component.³¹⁰ No patent will issue, not because there is no patentable subject matter, but because of lack of enablement.³¹¹

That the proposed test not be "under-inclusive" is equally important. As discussed above,³¹² the key technological advance in software as implementation inventions lies in the functionality they impart to the related special purpose computer system. Because the test eliminates the irrelevant machine readable media limitation it permits software as implementation to be claimed "per se,"³¹³ eliminating the possible wrongful exclusion of such inventions when claimed as such. Permitting claims to the invention in its actual form (rather than counter to the trend in industry away from hardware implementations) maintains the full incentive, free of irrelevant validity disputes and enforceable directly against anyone making, using or selling the software for use in its implementation role.³¹⁴

One final statutory subject matter argument remains: that convenient as it may be to treat appropriately limited claims to software im-

current knowledge to future implementations, including those which create new methods of programming or integrating software and hardware components or new hardware components themselves, to the extent permitted by the "doctrine of equivalents." See *supra* note 66 and accompanying text.

310. See discussion *supra* notes 303-04 and accompanying text.

311. Warmerdam's "bubble system" can be used demonstrate how the entire abstraction analysis works in the software context. *In re Warmerdam*, 33 F.3d 1354 (Fed. Cir. 1994). Merely claiming the general idea of creating a nested hierarchical bubble system, without more, fails to put the system to any technological arts use, whether or not the methodology for creating the bubble system is adequately enabled for § 112 purposes. All that is claimed is a good idea without any technological application. The application should be rejected for non-technological arts subject matter under § 101. If, however, a claim limited to a software implementation of the bubble system as part of a computer system would satisfy the useful arts first level of abstraction concern. The useful arts application is the computer system implementation itself, regardless of the specific application(s) of the computer system. However, such a claim must still be reviewed under § 112 to ensure that the computer system can actually be implemented by a person skilled in the art. If that is not the case, the invention only represents a conceptual computer software useful arts implementation of bubble systems. The application should be rejected, but in this instance, for lack of enablement under § 112.

312. See discussion *supra* notes 272-73, 283-89 and accompanying text.

313. See discussion *supra* notes 274-77 and accompanying text. Cf. K. Stephens & J. Sumner, *Patenting Software Objects and Aspects of Object-Oriented Programs under the New PTO Guidelines*, 23 AIPLA Q.J. 221, 235 (1995).

314. In a direct enforcement action a distributor may argue the software is being distributed exclusively for purposes other than for implementation as part of a computer system, thereby escaping through the claim limitation itself. The credibility of this argument will normally be extremely suspect: What is the alternative use? Why is the software distributed in (only) machine useable form? Why does the marketing indicate its use as a system component? Why are all the users actually using it in its implementation capacity? See discussion *infra* notes 343-45 and accompanying text.

plementation directly as articles of manufacture, the Constitutional and statutory subject matter provisions prevent this direct coverage of software functionality "as such" because of their intangible form.³¹⁵ There are three potential issues. First, as a general policy matter, is there a reason to impose such a limitation? The basic patentable subject matter inquiry is not related to tangibility but rather to whether the invention falls within the useful arts. Technological processes are "intangible" but pose no special subject matter concerns. At the policy level the only patentable subject matter issue is whether the useful arts limitation can be maintained for claims to intangible software implementation "as such." As discussed above, the proposal's requirement that the claims cover the software functionality only when used as instruction for a computer system provides precisely this assurance.

Second, one may be concerned that intangibility results in an ability to properly identify and test the invention for patentability. Concerns about intangibility should not be confused with patenting abstract ideas or § 112 definiteness concerns. The former are adequately dealt with by the implementation-language analysis as shown above. Concerning the latter, just as with process claims, we can ensure the metes and bounds of software implementation functionality are clearly defined and enabled by testing the description in the patent application itself.³¹⁶ Once a proper description is required under § 112, the remaining tests for patentability can be applied.

Finally, at a technical language level, there may be something in the Constitution or statute which prevents such coverage. There is nothing in the Constitution or patent statute that expressly requires tangibility of invention. In fact, as noted above, processes are expressly included. We may feel, however, that labeling software as implementation inventions, articles of manufacture or machines stretches those § 101 terms, which should be limited to things we can see and touch. In this regard we would do well to heed the Court's advice in *Chakrabarty* and *Diehr* that we not read more into the statutory requirements than is specifically dictated by the express language, especially when there is no policy justification for doing so. To read in a tangibility requirement confuses the ability to see or hold an invention with its existence.³¹⁷ There is

315. A further argument in defense of the need for computer readable media in the claims is that it provides a tangible fixation of the claimed software functionality which is required for patentability. See Stern, *supra* note 93 at n.12. The *Guidelines* inclusion of computer program listings "as such" (without computer readable media) in the non-patentable class of non-functional descriptive data arguably reflect this view. See discussion *supra* note 175 and accompanying text.

316. See discussion *supra* notes 65, 140-41, 308-11 and accompanying text.

317. Stobbs uses an excellent example involving a virus control program which operates entirely in cyberspace to demonstrate both the reality and technological application of that

certainly no requirement we be able to see inventions with the naked eye, although at an earlier date that would have likely conflicted with intuitive feelings about what constituted a machine or article of manufacture. As technology evolves, the coverage must be allowed to adapt so the policy net captures all matters falling within its intended reach, intangible or not.³¹⁸

At the end of the day any statutory language concern that does not reflect a underlying policy question should not a create a substantive barrier. If we believe that labeling intangible software implementation inventions articles of manufacture stretches the term beyond tolerance, we can simply call such inventions *per se* statutory processes³¹⁹ or otherwise amend the statute to expressly accommodate them.

b. Novelty and Nonobviousness

Merely because the proposed test ensures the patent incentive is properly applied to appropriate subject matter does not mean it results in satisfactory overall operation of the patent system. The test must also not impair, and if possible should enhance, the “progress” objectives embodied in the requirements of novelty under § 102 and nonobviousness in § 103.³²⁰

The proposed test’s separation of software as language claims from software as implementation also focuses and improves these inventiveness inquiries. Claims involving software as language are clearly identified by the test as implicating only the underlying idea being expressed. Therefore, they should be reviewed under §§ 102 and 103 in the same fashion as they are tested under § 101. No reliance may be placed on the

program notwithstanding its intangibility to humans. See STOBBS, *supra* note 18, Supplement at 40-41. See discussion *supra* note 153 and accompanying text (irrelevant whether visible to the human eye). A bit of this same issue has crept into the *State Street Bank* useful arts test for abstract idea application which requires there be a “useful, concrete and tangible result.” *State Street Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1373 (Fed. Cir. 1998). As the court ultimately found a mutual fund price to satisfy the test the “tangibility” portion of the requirement, tangible must not mean discernible by the human senses. It is not entirely clear, however, what it does mean. One possibility is that given the abstract idea context, the court was seeking a way to differentiate between merely theoretical notions and “things” having actual existence and interactive consequences.

318. For those needing further comfort, in the case of qualifying software patents the structure of the claims themselves, which requires the functionality be implemented as part of a computer system, ensures that in actual application the functionality will be manifested tangibility in electronic form. See *In re Lowry*, 32 F.3d 1579, 1583-54 (Fed. Cir. 1994) (citing *In re Bernhart*, 417 F.2d 1395 (C.C.P.A. 1969).

319. See Hollaar, *supra* note 268. This solution may, however, be less desirable because of the semantic confusion this may raise (revitalizing the software law of nature preemption arguments from *Benson* and *Flook*) and it’s possible effect on direct enforceability. See discussion *supra* notes 92-93 and accompanying text.

320. See discussion *supra* notes 47-49 and accompanying text.

software expression to support a finding of novelty or nonobviousness. The expressed idea must meet those requirements standing alone when measured against the prior art relevant to its (non-computer) field of application.³²¹

What remains is to ensure that software as implementation inventions are properly reviewed under §§ 102 and 103. For the system to work properly, this requires that a transition be made from the § 101 "invention as a whole" approach which supported the useful arts finding to determining and assessing the precise nature of the "inventive" aspects involved. This sounds, perhaps disturbingly, like the rejected § 101 point of novelty approach relocated to §§ 102 and 103. It is. As Professor Cohen has pointed out, although the dissection of an invention for § 101 purposes is inappropriate, breaking down an invention under § 102 and § 103 is perfectly proper, and in fact necessary, given the different policy objectives of these two sections.³²² For § 101 purposes, the issue is whether the claimed invention as a whole involves the useful arts. The inventiveness objective underlying novelty and nonobviousness focuses instead on what *specifically*, if anything, is new within the claimed invention. The former tests to ensure we are dealing with inventions in the useful arts; the second looks to whether the invention contains anything which represents progress in those same arts.

The easy software as implementation novelty cases, from an analytical perspective at least, are those in which the inventiveness exclusively relates either to the operation of computer system itself (increased speed or efficiency in the use of resources)³²³ or to the fact of automation (com-

321. If the novelty actually rests in the software implementation, then the invention should be claimed as software as implementation rather than as a broader software language claim covering the underlying functionality in general terms. See discussion *infra* notes 323-325 and accompanying text.

322. See, J. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 S. CAL. L. REV. 1092, 1168, 1172. See also Moy, *supra* note 242 and accompanying text (arguing that the PTO has been using a similar "blue pencil" rule for assessing novelty in hybrid inventions). Professor Cohen, finding "the absence of a principled basis under § 101 for separating statutory inventions from claimed inventions that are 'no more than' non-statutory mathematical algorithms is disturbing" relies on §§ 102 and 103 to eliminate undeserved patents. *Id.* at 1168. The proposed § 101 test in this article, coupled with a "point of novelty" approach under § 102 and § 103, provides such a principled basis for assessment of software related inventions under all three sections.

323. The subject matter of a number of the CAFC software cases involve precisely these types of inventions. They concern using the computer resources themselves in a novel way, whether through hardware directly or in software implementation. See, e.g., *Gottschalk v. Benson*, 409 U.S. 63 (1972) (fast internal conversion of BCD to binary); *In re Lowry*, 32 F.3d 1579 (data structure increasing speed and efficiency of operation); *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989) (fast multiplication without the need for expensive multiplier chips);

puterization) of the underlying activity.³²⁴ These innovations can fairly routinely be tested against the existing state of the computer arts using a "skilled in the computer arts" benchmark.³²⁵

The harder cases involve claims of novelty in the approach taken to underlying activity itself.³²⁶ Even assuming the applicant avoids § 101 problems by limiting her claims to coverage of the newly developed or discovered functionality in its software implementation as a part of a computer system,³²⁷ if the novelty is claimed to lie not in the computer activity but rather in the incorporated methodology then that methodology must be assessed to determine if it supports technological novelty and nonobviousness.

Two guideposts assist in keeping this assessment on track. First, properly applied, the test of novelty and nonobviousness is based on skill

In re Chatfield, 545 F.2d 152 (C.C.P.A. 1976), *cert. denied*, 434 U.S. 875 (1977) (parallel processing method).

324. These cases rely on the increased speed, efficiency or accuracy of putting the activity on a computer. *See, e.g., Iwahashi*, 888 F.2d at 1371-72; *In re Phillips*, 608 F.2d 879, 882 (C.C.P.A. 1979), *Chatfield*, 545 F.2d at 154. These claims will rarely succeed for the simple reason that there is nothing particularly novel or nonobvious to one skilled in the computer arts about using the computer as a tool to automate most activities to increase speed, accuracy or efficiency. In most cases, that is the inherent point of the computer automation. The European view of the unpatentability of software rests largely on this view. *See supra* note 278 and accompanying text.

325. The questions of availability of properly trained examiners and of the adequacy of databases to locate the appropriate prior art for performing this analysis, although improving, still remain. *See supra* note 52 and accompanying text. Note that it does not matter whether the increased speed or effective use of computer resources resides in the software, the hardware or both. Making that distinction irrelevant is precisely the point of the proposed test; any form of computer technology enhancement *per se* satisfies the useful arts requirement. If an invention goes beyond something particular to the specific application, because the computerization satisfies the patentable subject matter requirement without the need for further application, the applicant should consider claiming the coding technique more directly. This ability to claim innovative programming techniques generally goes a long way toward eliminating the need for *sui generis* protection of software. *See supra* note 285 and accompanying text.

326. The applicant can present this argument in pure terms or in combination with an "easy case" argument. The examiner should separate the two cases for analysis to avoid confusing a computer advance with a claim to improved methodology.

327. Although such a limitation would guarantee that the claims meet the requirements of § 101 under the proposed test, it will also restrict the scope of the patent to the computer implementation of the purportedly novel underlying methodology. If, however, the applicant expands the claims to eliminate the software implementation limitation (using the software as language to describe the underlying methodology itself) the software as implementation versus software as language analysis will require independent useful art assessment of the methodology under § 101. *See discussion supra* note 290 and accompanying text. Either way, an advance in a non-technological activity will not support a patent; in the former case under § 101 (directly as unpatentable subject matter) or under §§ 102, 103 102/103 (lack of progress of the useful arts); *see discussion infra* notes 330-331 and accompanying text.

in the art where the novelty resides.³²⁸ Therefore, the novelty and non-obviousness of the claimed underlying (non-computing) advance incorporated in the software must be measured against the knowledge of one skilled in that field of endeavor. This means the applicant and examiner must avoid mixing areas of expertise both to ensure identification of appropriate prior art and application of the proper level of expertise.

The point can be crystallized by examining claims to a computerized rubber curing system like the invention in *Diehr*. The software implementation of such a system is statutory subject matter if the claim coverage is limited to the software functionality when serving as a component of the computer driven curing system. Assuming it has thus cleared § 101, the novelty/non-obviousness analysis runs as follows:

If the novelty lies in the operation of the system as a computer system or in having computerized the activity, then the underlying curing process is taken as a given (deemed existing prior art). The novelty of the computer system implementation can then be assessed, as an easy case as discussed above, for increased speed, efficient use of computer resources or the idea to automate the process. If, however, the claimed novelty lies in the rubber curing process itself, then the computerization aspects are taken as given³²⁹ and the curing methodology itself must be tested against those of ordinary skill in rubber curing arts. Unless the invention involves a new insight in that field, the novelty test will remain unsatisfied.

In neither case should the focus cross between the two areas of activity, leading to a finding of novelty or lack of obviousness based on lack of relevant prior art or skill in the other unrelated field.³³⁰ In short, it is irrelevant whether an expert in rubber curing could create the computer software implementation or a computer expert could come up with the rubber curing methodology.

The second guidepost is that novelty and nonobviousness cannot be predicated on advances outside the technological arts.³³¹ This is because properly understood, the novelty test does not merely require that an

328. The *Guidelines* recognize and properly reflect this point. See *Guidelines*, *supra* note 4, at 7486-87.

329. See Cohen, *supra* note 322, at 1169 (citing to Professor Stern's "innovative programmer standard" from R. Stern, *Tales from the Algorithm War: Benson to Iwahashi, It's Deja Vu All Over Again*, 18 AIPLA Q.J. 371 (1991)). If, in contrast, the claimed novelty is in the computerization technique, the advance should be tested against what is known in that art; not by assuming everything in the claim is prior art. *Id.* at 1174; see discussion *supra* notes 323-25 and accompanying text.

330. Finding the "point of novelty" may be somewhat problematic but is hardly impossible. See discussion *infra* notes 314-316 and accompanying text.

331. See Cohen, *supra* note 322, at 1171 ("A claimed invention that, taken as a whole, is (or may be) statutory subject matter is nonetheless unpatentable if its novelty and nonobviousness inheres in its non-statutory elements."); Moy, *supra* note 242.

advance of any kind has occurred, but that patents only be issued in connection with progress in the technological arts. To grant a patent for a computer software invention in which the only novelty resides in an advance in the non-technological arts would misapply the patent laws incentive to progress outside the proper scope of the system.

The easy case is straight-forward enough. If there is agreement that the only novelty is found in a new methodology in a non-technological field, *e.g.*, a new law of nature, method of accounting or form of political organization (as distinguished from the computer techniques or computerization itself), then the invention fails for lack of novelty under § 102.

The more difficult case involves a simultaneous development of what is admittedly a non-technological advance and the software technology used to implement that development. On its face this situation appears to result in a technological advance. The truth is otherwise. The acceleration calculator example from above demonstrates the point. Assume the inventor is Sir Isaac who in 1998 has simultaneously discovered the famous equation and filed a patent application on a software driven acceleration calculator. Assume further there is a complete absence of acceleration calculators so that Sir Issac may certainly claim technological novelty in the absolute sense; the prior art clearly demonstrates his is the first acceleration calculator.

The novelty inquiry must, however, be pressed more deeply to ascertain the source of the novelty if the patent policy objectives are to be respected. If it turns out that had others known the formula, the necessary computing technology was not available until invented by Sir Issac, and therefore, the technological novelty in the computer implementation will sustain limited claims to a software as implementation invention under § 102. These claims, however, will only extend to the novel computerization of the formula, not to the formula itself.

If, as is more likely the case, the invention employs exclusively existing computerization technology and the dearth of acceleration calculators is due exclusively to the fact that $F=ma$ relationship has just been revealed, no patent should issue. The claim to novelty in this latter situation is in reality only to the new formula, so the source of the advance falls entirely outside the technological arts. To grant a patent on the acceleration calculator in these circumstances, even though limited to the useful arts application, would promote only progress in the sciences and not in the technological arts.³³² Although the application would

332. This means any source of novelty falling outside the technological arts, such as a new method of doing business (moving this former subject matter inquiry to the less problematic § 102 and § 103 context) or the discovery and statement of a new law of nature will be insufficient. The ability to exclude newly developed mathematical systems or models is

meet the § 101 requirement, it would fail under § 102 and no patent should issue.

This analysis re-raises the possibility that the more proper analysis of these situations belongs under § 101, as Justice Stevens indicated in *Flook*: a claim to scientific novelty is simply to the “wrong stuff” for a patent.³³³ The argument, however, misses the difference between the patentable subject matter and novelty inquiries just as the “point of novelty” approach did in *Flook*. A limited claim drafted to cover only the computer software implementation (the driver for a computerized acceleration calculator) of the scientific principle is technology and proper subject matter. The fault lies solely in the fact that there is no new technology in the acceleration calculator software implementation for § 102 purposes.³³⁴ The discovery of the scientific formula did not result in

particularly important in the computer software field. Such systems or models, which are used to express general relationships independent of any specific content of the system symbols employed, may form the basis for faster or more efficient computer systems. See, e.g., *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989) (using a mathematical model for squaring which replaces the multiplication operation; in that case the model was pre-existing). Although arguments can be made that such new mathematical models should be patentable, at present they fall under the “laws of nature” or “abstract ideas” rubric, depending on one’s proclivities (see *supra* note 303 and accompanying text) and, standing alone, are not patentable. See *Karjala*, *supra* note 285, at n. 38; note 221. Similarly, nor does their discovery alone support a finding of technological novelty in a software as implementation invention, although recognizing the benefits of their computerization might, if non-obvious to one of ordinary skill in the art. With regard to laws of nature, there is also the technical argument, used in connection with § 101, but equally applicable to this discussion, that they have always been present and rather than having been newly invented, have only been newly discovered. See *Hollaar*, *supra* note 268, at 299 (“they have no inventor”); Cf., *Parker v. Flook*, 437 U.S. 584, 593 (1978) (“laws of nature are not new”); *In re Sarkar*, 588 F.2d 1330, 1333 (C.C.P.A. 1978) (making the same point in the § 101 context).

333. See discussion *supra* notes 59-60 and accompanying text; see discussion *supra* note 335 and accompanying text.

334. Addressing the novelty inquiry in this fashion may not technically match up with the language of § 102 which contains specifically articulated “prior art” type novelty requirements. 35 U.S.C. § 102 (1994). Although one could argue there simply has been no “invention” to get this novelty argument within the literal language of § 102, this sounds too much like the properly discredited “point of novelty” approach to patentable subject matter. The better argument may be that just as there is a separate “usefulness” requirement under § 101, there is also a distinct “novelty” requirement under the “new and useful” language. Both tests are independent of the statutory subject matter test, with “new” and “useful” modifying the listed statutory classes of process, machine, article and composition of matter. A claim to a newly discovered law of nature or newly developed method of doing business, standing alone, is non-statutory subject matter because it does not fall within the specified classes of useful arts. A machine (e.g., a computer system) in which the sole novelty comes from incorporating that law of nature or method of doing business will be statutory, but is not “new” because the novelty does not progress the useful arts. If under *Diehr*, § 102 is as amplification of the § 101 “new” requirement this textual argument fits. See *Diamond v. Diehr*, 450 U.S. 175, 189-191 (1981) (the majority in attacking the point of novelty approach, notes the Congressional reports indicate that § 102 is intended as “as

novel software implementation techniques, just old software implementation of newly discovered laws of nature. The invention results in no *progress* in the useful arts.³³⁵

As a practical matter, moving the point of novelty analysis to § 102 and § 103 will still involve complex identification issues. This effort, however, is significantly more straight-forward than seeking to disqualify applications based on unpatentable subject matter by hunting for hidden preemption of laws of nature, methods of doing business or other non-technological activities in claims which on their face, when viewed as a whole, are clearly in the useful arts. Instead of an exercise in classification, the issue is reduced to identifying where the applicant claims the novelty to lie.³³⁶ Although this still can be a difficult task, the exam-

amplification and definition" of "new" in § 101). If, however, § 102 is meant as an exclusive definition of the word "new" in § 101, the argument fails.

335. Justice Stevens' misplaced § 101 "point of novelty" invention parsing approach in *Flook* and in dissent in *Diehr*, is right on point when applied to § 102. The articulation of the argument in his dissent in *Diehr* demonstrates the point: "proper analysis, therefore, must start with an understanding of what the inventor claims to have discovered—or phrased somewhat differently—what he considers his inventive concept to be." See *Diamond v. Diehr*, 450 U.S. 175, 212 (1981) (Stevens, J., dissenting). This is not a patentable subject matter question but an inquiry into where the inventor feels the novelty to lie. If the software claim is properly stated under the implementation test it will cover the computer software component of a special computer system. That is patentable subject matter. For purposes of novelty, however, one must then look, as Justice Stevens' suggests, to specifically where the inventive activity lies to determine if this patentable subject matter invention has progressed the useful arts. Justice Stevens' own example in *Flook* concerning the Pythagorean Theorem is instructive. See *Parker v. Flook*, 437 U.S. 584, 590 (1978). Standing alone the Theorem is not technology and a claim to it "as such" whether expressed in software or otherwise is not patentable subject matter. If, however, the claim is to a software program which forms a part of a computerized surveying apparatus the analysis changes. A claim covering the computerized apparatus (or just the software component if properly limited) is statutory subject matter. If, however, the computerized surveying apparatus (or the software component) makes no advance in the field of computerized surveying apparatus (of computing) other than to incorporate the newly discovered Theorem and implement its calculation, the invention has contributed nothing by way of technological advance. It should not be considered novel under § 102.

336. Many of the same tactics used by applicants in the FWA context will likely reappear in the novelty inquiry. Fortunately, the change in the nature of the inquiry substantially simplifies the process, both substantively and administratively. For example, applicants may seek to rely, as they have to satisfy the FWA test, on data-gathering or output activities as a source of technological novelty. See *supra* note 90 and accompanying text. Under the implementation versus language test the inquiry is relatively direct. The invention is or is not statutory subject matter under § 101 based on the component of a computer system limitation. There is no need to determine if the input or output activities constitute a sufficient implementation of a non-technological methodology or a law of nature. Under § 102, as the applicant claims the novelty lies in the input or output activity, it is those specific activities which are tested directly against the relevant prior input-output art without reference to the unpatentable methodology or law of nature and will constitute the key claims limitations in a patent which actually issues.

iner can press the issue by inquiry until a sufficiently definitive answer is obtained.³³⁷ Ultimately the examiner can indicate she will simply assume that the underlying non-computing methodology lies in the prior art, will be taken as given for novelty purposes, and the computerization will be examined for novelty, unless demonstrated otherwise by the applicant.³³⁸

Applying these two guideposts substantially clarifies the novelty inquiry in the context of software related patents. Focusing on and limiting the specific source of novelty in such inventions ensures that mere auto-

337. In many software cases the analysis will fall out relatively cleanly. If the applicant specifically claims the novelty lies in a particular facet of the computer operation that can be test directly against the relevant art. If there is no novelty, then the invention fails the § 102 test. In some cases, however, determining whether the novelty lies in the computer aspects or the non-technological aspects may be unclear. To ensure that there is no improper reliance, or improper tainting, additional effort will be required to explore the novelty issue in these situation. Because the inquiry is focused on determining where the novelty lies, the examiner can push the applicant in that specific direction. For example, the applicant claiming the computerized rubber curing system (discussed *supra* note 329 and accompanying text) may generally claim the novelty lies in the new ability to continuously monitor the temperature and automatically open the mold. The examiner can press to determine if that novelty lies in programming a computer to continuously monitor and automatically open, or in recognizing the methodological value of continuous monitoring and automatic opening. If it is the former, then the programming is tested against the state of the computer arts. If it is the latter the methodological improvements should be tested against the rubber curing arts. If it is in both, each is tested only against the relevant prior art.

Professor Cohen provides an interesting additional example. In *Arrhythmia Research Technology v. Corazonix*, 958 F.2d 1053 (Fed. Cir. 1992) a case of useful arts advance in the method for analyzing EKG data using a computer system or is the novelty "the discovery that certain naturally occurring EKG patterns are correlated with a likelihood of later heart failure." See Cohen, *supra* note 322 at 1157-58. The question for the examiner and applicant to resolve for purposes of § 102 is whether there is any new technology in the invention. This novelty can be found in either the computing techniques or in the ability of the computer system to identify and extract the naturally occurring EKG patterns. If, however, the only novelty rests in having identified the existence of the EKG patterns and the system merely applies existing computing solutions to automating their collection and processing, § 102 will bar the issuance of a patent.

This type of novelty inquiry should also not penalize "smart" inventors. As the courts have noted, it would be ironic to grant patents to inventors who only understand practically how to create an apparatus and, therefore, insist the novelty is in the implementation, and deny a patent to those which can also explain why the implementation works (by explaining the new methodology or law of nature on which it is founded). See, e.g. *In re Bernhart*, 417 F.2d 1395, 1399-1400 (C.C.P.A. 1969) (addressing the issue under § 101). In these cases, the examiner will need to pay special attention to ensuring that the novelty truly lies in the implementation by focusing carefully on what variations have been made in the prior art and why. See discussion *supra* note 332 and accompanying text (discussing simultaneous discovery of a law of nature and creation of the implementation technology).

338. This seems reasonable in light of the fact that the claim is to software as implementation. If there is additional novelty, the applicant is in the best position to identify it.

mation or use of standard programming techniques will be insufficient to obtain a patent and that such claims cannot serve as a vehicle for capturing non-technological methodological building blocks.

c. Administrative Ramifications

In addition to achieving the desired policy objectives, a suitable § 101 test will promote efficiency, consistency and predictability in patent practice and administration.³³⁹ On all three counts, the proposed test succeeds by virtue of its simplicity. It is straight-forward in its sole requirement: the claim must only cover the software functionality when it serves as the instructional component of a computer system implementation. Certainly, in the grand tradition of patent law, creative draftspersons will seek to push the boundaries in a variety of ways. At the end of the day, however, the examiner's § 101 inquiry is only whether the claims read on uses of the functionality beyond the incorporation of the software into a computer system. Ensuring this limitation is accomplished in each case, although technically important, is not a complex task.³⁴⁰

339. See *e.g.*, *Diamond v. Diehr*, 450 U.S. 175, 219 (1981) (Stevens, J., dissenting).

340. Approaches could include regulations or certain formulations of preamble which clearly imposes the limitation. See Stern, *supra* note 93; see *infra* Appendix. If properly pursued by the examiner, the file wrapper should always demonstrate the intention to create the limitation.

The question of whether the claims actually limit use to software implementation is the key issue in many of the CAFC cases. For example, in *Trovato* the process claims involved a method for determining "a least cost path." See, *In re Trovato*, 42 F.3d 1376, 1377 (Fed. Cir. 1994), *vac. and remanded*, 60 F.3d 807 (Fed. Cir. 1995) (en banc). The related apparatus claims are to "Apparatus for planning a least cost path comprising: means for . . . means for . . ." where the "means for" describe the least costs path method of the process claims. The disclosure in the specification was flow charts and computer code. *Id.* at 1377-79. The original CAFC panel in *Trovato* found the process claims unpatentable as "nothing more than a process of performing a numerical calculation." *Id.* at 1380. They also found the apparatus claims unpatentable because "they do not disclose a machine of any sort. . . [A]ll the disclosed means are simply software instructions; no 'structure' appears in the specification." *Id.* at 1382. The implementation versus language test would run as follows: the process claims are an easy case. They stand on their own without consideration of any software context. Without more, least cost path methods are not technology, are at best economic modeling. On the other hand, an apparatus claim expressed in software terms must be analyzed for the computer system component limitation. An invention containing that express limitation is certainly statutory subject matter under § 101, although it may or may not be patentable subject to examination for enablement under § 112 and novelty and nonobviousness under §§ 102 and 103. The question in *Trovato* is whether the use of the software was, in fact, so limited. If the preamble limitation or the § 112, paragraph 6 analysis of the "means for" equivalents are insufficient, as the original panel found, then there is no apparatus and no patentable subject matter. *Id.* at 1382 (contrast this with the finding in *Alappat* on the preamble issue, see *supra* note 256 and accompanying text. If, however, the record shows either the limitation or equivalents clearly limit coverage of the methodology to a software implementation, then the apparatus

This framework also results in clarification of a number of other interactions and relationships in the examination process: the prior art field of specialty must match that of the claimed novelty; reliance on non-technological arts advances in a novelty inquiry are not enough under § 102; and the enablement role of § 112 is separated from abstract ideas in the § 101 inquiry.

Finally, because the nature of the invention will be clearly stated as a result of the § 101 examination, it is either about software as implementation (computerization) or about the express and not the software. By utilizing a point of novelty approach under § 102 and § 103, the appropriate inquiry is much more clearly focused both for the applicant and for the examiner.

d. Other Considerations

It would be remiss not to note that other ramifications of permitting patents on software *per se* when limited to its role as a machine component. Patents in this form may, for example, raise issues concerning marking (what is marked if no computer readable medium is required) and lack of associated notice or how first sale concepts should be applied (are they process claims or apparatus claims for first sale purposes).³⁴¹ The key policy objective, however, is to ensure inventions are correctly classified and reviewed for patent incentive purposes. Having accomplished this task, the remaining fine-tuning can be adjusted to suit the result.

e. Overall Effect on Software Patentability Outcomes

As a consequence of clarifying the § 101 tests and tightening up the § 102 and § 103 review, the model as a whole will likely result in fewer grants of software related patents. Although the implementation versus language model marks out a clear article of manufacture path over the § 101 hurdle, the route is only open to those willing to limit their claim coverage to software used in computer implementation. All other claims, whether or not couched in software terms, will be treated as software as

claims should satisfy § 101 even absent hardware or other physical structure. This analysis means only that § 101 is satisfied, not that a patent will issue. There may remain, for example, serious novelty issues. See *supra* note 301 and accompanying text. Similar analysis can be applied to the bid maximization method at issue in *In re Schrader*, 22 F.3d 290 (Fed. Cir. 1994); the auto-correlation unit in *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989); the sales management tool in *In re Maucorps*, 609 F.2d 481 (C.C.P.A. 1979); and the translation program in *In re Toma*, 575 F.2d 872 (C.C.P.A. 1978).

341. Without appropriate categorization, permitting applicant's to draft around the requirements of the Patent Act can have undesirable policy side-effects. See Thomas, *supra* note 14. However, once the basic categorization tests are sound, the additional objectives can be confidently attached to that structure.

language patents subject to § 101 review based exclusively on the underlying functionality being expressed.

If the § 101 barrier has been lowered (or at least clarified and simplified) for software as implementation cases, the §§ 102 and 103 requirements have been raised. A specific inquiry into the actual source of novelty will quickly eliminate any claims to routine computing solutions or claims based on non-technological advances. This is entirely consistent with the policy objectives of the patent regime. The result is only truly new and nonobvious advances in computer technology or in methods of computerizing activities will be recognized by the grant of a patent on a software invention.

IV. CONCLUSION

The proposed software as language versus software as implementation model provides an efficient and effective method for determining the patentability of software related inventions. Using this framework helps ensure that inventions claimed in software terms will result in the issuance of patents which meet the full reach of the useful arts policy incentive underlying the patent laws, but no further. In addition, those patents which do issue will be free of artificial constraints on the structure of their claims or the method of their enforcement, permitting us to realize the full value of the patent incentive system.

APPENDIX

SOME PRACTICAL CLAIMING RAMIFICATIONS

I. Several sample software as article of manufacture claim formats are listed below:³⁴²

1. An article of manufacture, comprising:
a computer readable medium bearing computer program code embodied therein for performing a task and including:
 means for ;
 etc. . . .
2. A computer program product, comprising:
a computer readable medium bearing computer program code for performing a task and including;
 means for '
 etc. . . .
3. A computer readable medium having stored thereon computer software instructions for execution by a central processing unit to cause a computer system to perform the steps of:
 executing a first task;
 executing a second task;
 etc. . . .

ANALYSIS UNDER THE PROPOSED IMPLEMENTATION-LANGUAGE TEST

Claim formats 1 and 2 contain strong implications of the "implementation" limitation, however, Claim 3 is the closest to meeting the requirement that the claims only cover the software functionality when used as a component of a computer system.

Another, clearer alternative might be: "A computer software program product used as a component of a computer system, consisting of instructions for execution by a computer processor which cause the computer system of which the processor is a part to perform the steps of:"

As Jeff Draeger points out in his paper, care must be taken not to

342. See Becker, *supra* note 191 at 16 (1996). This article also contains a number of interesting additional format variations relating to claiming computer software.

draw the claim limitation too narrowly.³⁴³ If the claim language states that the software is *exclusively* for use in a computer system, an infringer who creates "dual use" software (e.g., for both execution and for reading) may technically escape the coverage of the claims.³⁴⁴ A claim to software "used as a component" is, therefore, preferable. This claim language still limits the patent's coverage to the software functionality when used as a machine component (implementation), thus eliminating any concern that the claim implicates software as language. Third parties remain free to make, use or sell other applications of the functionality. However, any third party making, using and selling the software functionality for machine component use will be caught by the claims, whether or not it also has alternative uses.³⁴⁵

II. Below are several of the article of manufacture claim formats which were at issue in *Beauregard*.

1. An article of manufacture comprising:

a computer usable medium having computer readable program code means embodied therein for causing something to happen, the computer readable program code means in said article of manufacture comprising:

computer readable program code means for causing step one;

343. See, Jeffrey S. Draeger, *Are Beauregard's Claims Really Valid*, 17 J. Marshall J. Computer & Info. L. 347 (1998).

344. *Id.*

345. The third party is only liable, of course, to the extent of their infringing use. They thus may claim that their product is sold exclusively for the non-infringing, non-implementation use of communicating the underlying algorithm to the reader. This defense may raise issues similar to those found in the contributory infringement inquiry, with the outcome turning on whether there is intent to pursue the infringing use and if there actually are substantial, non-infringing uses. See *supra* note 92 and accompanying text. In practice, because almost all infringement cases will be brought precisely because the software is actually being made and distributed for use as a machine component the defendant will be expressly marketing the product as a computer system component (software for use in a computer system). Additionally, users who are in fact acquiring the product for computer implementation will insist the product be delivered in computer executable form which will have little communicative potential. Therefore, it will be unlikely that a defendant will prevail. As noted in the article, as technology evolves these practical difficulties to a non-infringement defense may eventually disappear. See discussion *supra* note 275 and accompanying text. This will focus the inquiry more firmly on the question of intent and alternative uses. In all events, if the end-user colludes with their supplier, both can be joined in the infringement action.

computer readable program code means for causing step two;

etc. . . .

2. A computer program product for use with a graphics display device, said computer program product comprising:

a computer usable medium having computer readable program code means embodied in said medium for causing . . .

3. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps causing something to happen, said method steps comprising:

task 1,

task 2,

etc. . . .

