

UIC John Marshall Journal of Information Technology & Privacy Law

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

Article 10

Fall 1998

Are Beauregard's Claims Really Valid?, 17 J. Marshall J. Computer & Info. L. 347 (1998)

Jeffrey S. Draeger

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



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

Recommended Citation

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

<https://repository.law.uic.edu/jitpl/vol17/iss1/10>

This Comments is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in UIC 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.

COMMENT

ARE BEAUREGARD'S CLAIMS REALLY VALID?

I. INTRODUCTION

Computer hardware, software, and networking equipment together have fueled the onset of the Information Age, an age where a seemingly endless stream of ones and zeroes often is an extremely valuable commodity. Undeniably, computer software has been a crucial building block in this Information Age. However, intellectual property law has been slow to embrace software inventions as patentable subject matter on par with computer hardware and networking equipment.

A computer executing software is now judicially recognized as being a special purpose computer, which may be patentable due to the steps performed by the software when executed by the computer. Unfortunately, absent the machine, courts have not been willing to validate the patentability of claims to computer instruction embodied in a computer readable medium. Such claims are likely the most useful type of software claims in an enforcement scenario because the direct infringer makes or sells the allegedly infringing software. Other types of claims, such as machine or process claims, typically require execution of the software, and a litigant must resort to pursuing end-users or relying on less desirable contributory or induced infringement theories.

This Comment examines the technological and legal basis for claims to software embodied in a computer readable medium. Despite being endorsed by the "Examination Guidelines for Computer-Related Inventions" ("*Guidelines*")¹ published by the United States Patent and Trademark Office ("PTO"), such claims lack a solid judicial foundation.

To understand the context in which the *Guidelines* arose, this Comment begins by analyzing the decisions that led to the *Guidelines*, including a number of cases which apply to software patents in general. The analysis focuses on software embodied in a computer readable medium. It demonstrates that the patenting of computer instruction embodied in

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

a computer readable medium, as permitted by the *Guidelines*, is a clear extension beyond the precedent of the Court of Appeals for the Federal Circuit ("Federal Circuit") and the United States Supreme Court. The Comment also gives a technical perspective on why software, so long as it is embodied in some product, should be treated no differently than hardware, and, consequently, why computer instruction embodied in a computer readable medium technically qualifies as patentable subject matter. Furthermore, the Comment demonstrates why the rationale of *In re Lowry*² is not conclusive on the issue and provides an alternative technical and legal foundation upon which such claims can be upheld.

Since the Federal Circuit's precedent does not cover all types of claims endorsed by the *Guidelines*, this Comment explores claim language, which most closely tracks judicially approved claim language. Thus, the analysis applies the technical and legal concerns of claims to software embodied in a computer readable medium to claim drafting. The analysis further discusses the potential dangers of claiming in "step plus function" format, in light of recent Federal Circuit case law, and provides recommended claim formats which reduce the likelihood of claims being found invalid or being construed in an unduly narrow manner. Finally, the Comment briefly discusses other hurdles to enforceability of software patents drawn to computer readable media. Specifically, the analysis addresses the enablement, description, definiteness, and best mode requirements of 35 U.S.C. § 112.

This Comment concludes that claims for computer instruction embodied in a computer readable medium *do* constitute statutory subject matter. As proper statutory subject matter, such claims should be evaluated "as a whole," including the claimed functions embodied in the computer instruction. The discussion in the Comment regarding claim and specification drafting should assist a practitioner in obtaining broad patent protection in view of existing judicial precedent.

II. BACKGROUND

THE THORNY ROAD TO JUDICIAL RECOGNITION OF COMPUTER RELATED INVENTIONS: OVERCOMING THE NOTION THAT COMPUTERS PREEMPT THE USE OF ALGORITHMS

The legal community has at least embraced the technological reality that a computer system executing software constitutes a patentable machine. The courts have specifically acknowledged that a "general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions

2. *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994) (discussed extensively, *infra*).

from program software.³ Similarly, computer executed processes have long been recognized as patentable.⁴ Even though these tenets of patent law are now well established, a brief overview of the historical developments is useful in understanding the state of the law prior to the adoption of the *Guidelines*. With this understanding, the practitioner can see the differences between the adopted *Guidelines* and the existing state of the law at the time of adoption.

The first notable case relating to computer software is *Gottschalk v. Benson*,⁵ in which the Supreme Court first considered the patentability of a computer program. The patent application involved a method of programming a general purpose digital computer to convert signals from binary coded decimal ("BCD") form into pure binary form. The Supreme Court held the claims unpatentable under 35 U.S.C. § 101.⁶ The Court noted that the application attempted to claim a mathematical formula and the analytical steps involved in solving the formula.⁷ The Court pointed out that the claims were "not limited to any particular art" and that conversion of BCD numerals to pure binary numerals "can be done mentally."⁸

The Court found that as a general rule "phenomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work."⁹ The Court stated that "[i]f there is to be invention from such a discovery, it must come from the application of the law of nature to a new and useful end,"¹⁰ and concluded that the claims in question would have the "practical effect" of wholly preempting any use of the mathematical steps in the conversion formula.¹¹

The Court expressly stated in *Benson* that its holding should not be construed as excluding computer programs from patentable subject matter:

It is said that the decision precludes a patent for any program servicing a computer. We do not so hold [W]hat we come down to in a nutshell is the following. It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting BCD numerals to pure binary numerals were patented in

3. *In re Alappat*, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (quoting *In re Freeman*, 573 F.2d 1237, 1247 n.11 (C.C.P.A. 1978)).

4. *See, e.g., Diamond v. Diehr*, 450 U.S. 175 (1981).

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

6. *Id.* at 71.

7. *Id.* at 66.

8. *Id.*

9. *Id.* at 67.

10. *Id.*

11. *Id.* at 71.

this case.¹²

Despite this express acknowledgment, *Benson* had the consequence of suggesting that computer programs are unpatentable.

The apparent unpatentability of software was further suggested by the Supreme Court's decision in *Parker v. Flook*.¹³ The *Flook* case analyzed a method claim for updating an alarm limit of at least one variable involved in a process for the catalytic conversion of hydrocarbons. The sole novelty in *Flook's* claims resided in the mathematical algorithm for calculating the updated alarm limit. The Court found that the simple post-solution activity of updating the alarm limit could not save the process.¹⁴ The Court emphasized in *Flook* that the mere fact that a claim may contain an algorithm does not render the entire claim unpatentable.¹⁵ The Court also noted that claims must be considered as a whole,¹⁶ a point later reiterated in *Diamond v. Diehr*.¹⁷

Although the Supreme Court admonished that the *Benson* and *Flook* decisions did not deny protection to all computer programs, *Diehr* was the first decision of the Supreme Court specifically holding a computer process to be statutory subject matter within the definition of § 101. In *Diehr*, the claimed invention related to a "method of operating a rubber-molding press" by using the well known Arrhenius equation to control the cure time of synthetic rubber. The Court honed in on that which it regards as non-statutory subject matter:

Excluded from such patent protection are laws of nature, physical phenomena and abstract ideas (citations omitted). An idea of itself is not patentable. A principle, in the abstract, is a fundamental truth; an original cause, a motive; these cannot be patented, as no one can claim in either of them an exclusive right.¹⁸

The Court stated, however, that an application of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection, and found *Diehr's* process to be such an application.¹⁹

Following *Benson*, *Flook*, and *Diehr*, a series of cases focused on whether the claims preempt a mathematical algorithm. "Mathematical algorithms" were excluded on the basis that they fell within *Diehr's* prohibition against patenting laws of nature. A two step test called the Freeman-Walter-Abele Test developed; however, this test was soon criti-

12. *Id.*

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

14. *Id.* at 590.

15. *Id.* at 591.

16. *Id.* at 594.

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

18. *Id.* at 185.

19. *Id.* at 187-88.

cized as not being the only test for making § 101 determinations.²⁰ The test may be stated as follows:

It is first determined whether a mathematical algorithm is recited directly or indirectly in the claim. If so, it is next determined whether the claimed invention as a whole is no more than the algorithm itself; that is, whether the claim is directed to a mathematical algorithm that is not applied to or limited by physical elements or process steps. Such claims are nonstatutory. However, when the mathematical algorithm is applied to one or more elements of an otherwise statutory process claim, the requirements of § 101 are met.²¹

The Freeman-Walter-Abele Test may be little more than a useful analytical tool after *In re Alappat*.²² In *In re Alappat*, the Federal Circuit (sitting en banc) set forth an alternative analysis. The Court stated that § 101 determinations must be made in accordance with the language of § 101 and the Supreme Court decisions in *Benson*, *Flook*, and *Diehr*. With respect to claims containing algorithms, the Federal Circuit explained that these cases demand that the "claims must be considered as a whole."²³ The *In re Alappat* court conveyed the idea that certain types of mathematical subject matter represent nothing more than abstract ideas (such as mathematical algorithms) until reduced to some type of practical application. If the practical application is claimed, the presence of a mathematical algorithm does not bar patentability.

Notably, the claims at issue in *In re Alappat* were drafted in means plus function format implicating the actual computer hardware described in the specification under § 112. Thus, the Court passed judgment on Alappat's invention by considering the structure in the specification as a part of the claims. Under the "claims as a whole" approach, the court was required to consider a machine configured by its software to be a special purpose computer. Consequently, the "claims as a whole" mandate would also appear to require consideration of computer process steps claimed as part of a computer product claim.

Thus, the *In re Alappat* decision further opened the door for arguing that data structures and computer programs, when embodied in a computer product, were indeed statutory subject matter. Whether process, machine, or manufacture, *In re Alappat* indicates that the underlying processes in claims to software for a general purpose computer are to be evaluated by the limitations recited in the claims rather than esoteric concerns about preempting algorithms.

20. See, e.g., *In re Meyer*, 688 F.2d 789, 796 (C.C.P.A. 1982).

21. *In re Schrader*, 22 F.3d 290, 292 (Fed. Cir. 1994) (quoting *Arrhythmia Research v. Corazonix*, 958 F.2d 1053, 1058 (Fed. Cir. 1992)).

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

23. *Id.* at 1557.

III. ANALYSIS

COMPUTER INSTRUCTION EMBODIED IN A COMPUTER READABLE MEDIUM
LEAPS OVER THE § 101 HURDLEA. DO DATA STRUCTURES AND PROGRAMS IN MEMORY QUALIFY AS
A "MANUFACTURE?"

As seen in the previous discussion, computer processes and general purpose computers executing programs have gained acceptance as patentable subject matter. Patenting software stored on a computer readable medium (or in memory) by itself, however, is a bit more controversial. A computer disk itself is functionally the same as another disk with different contents until it cooperates with a computer. Therefore, it is crucial that the contents of the disk be considered in novelty and non-obviousness determinations. However, whether information programmed on a computer readable medium really adds functional characteristics to a manufacture, which by itself remains inert, may be questionable.

1. *Defining a "Manufacture"*

In evaluating the patentability of a "manufacture" storing computer instruction, it may be useful to first look to the statute for guidance. § 101 of Title 35, United States Code, provides "[w]hoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."²⁴

Unfortunately, Title 35 fails to provide any further definition of "manufacture;" however, a fairly expansive interpretation has emerged. A "manufacture" is generally regarded as a class of inventions bounded by machines on one side and compositions of matter on the other.²⁵ In other words, the term "manufacture" contemplates the class of all man-made items "not found in substantially the same form in nature that are neither machines nor compositions of matter."²⁶

The leading Supreme Court case on whether an object is a manufacture, *American Fruit Growers, Inc. v. Brogdex Co.*,²⁷ held that an orange impregnated with borax was not a patentable article of manufacture.²⁸ The court quoted a dictionary definition of manufacture which hinged on whether the materials combined in the manufacture were given new

24. 35 U.S.C. § 101 (1994).

25. 1 DONALD CHISUM, CHISUM ON PATENTS § 1.02[3] (1998).

26. CHISUM, *supra* note 25, at § 1.14.

27. *American Fruit Growers, Inc. v. Brogdex Co.*, 283 U.S. 1 (1931).

28. *Id.* at 11.

forms, qualities, properties, or combinations.²⁹ The borax impregnated orange was found to be suitable only to previously known uses, and therefore, not patentable as an article of manufacture.³⁰

On the other hand, in *Diamond v. Chakrabarty*,³¹ the Supreme Court upheld a broad concept of a manufacture. In *Chakrabarty*, the court held a genetically-altered living microorganism to qualify as patentable subject matter.³² The court cited the legislative history of the 1952 Patent Act which suggested that Congress intended that "anything under the sun that is made by man" be included in patentable subject matter.³³

Programming a computer readable medium with computer instructions places the computer readable medium within the broad definition of manufacture of *Brogdex* and *Chakrabarty*. The "material" (*i.e.*, the disk) is given new "qualities" and "properties" in that the programmed disk causes the computer to function in a different manner. For example, a compact disk programmed with the video game "Mortal Kombat"³⁴ surely is given new "qualities" and "properties" by the placement of the program thereon. Namely, it allows a user to turn a computer into a virtual stage for a bloody massacre. On the other hand, the disk does not function any differently *by itself*. This disk by itself obtains no new utility, but rather, is by itself a non-functional sub-component. A programmed compact disk and a blank compact disk, by themselves, are equally useful as little more than Frisbees.

Relevant to this non-functional sub-component aspect of computer readable media are cases relating to kits and parts of kits, since parts of

29. *Id.*

30. *Id.* at 12.

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

32. *Id.* at 309.

33. *Id.* at 308-309. The *Chakrabarty* Court stated:

In choosing such expansive terms as "manufacture" and "composition of matter," modified by the comprehensive "any," Congress plainly contemplated that the patent laws would be given wide scope. The relevant legislative history also supports a broad construction. The Patent Act of 1793, authored by Thomas Jefferson, defined statutory subject matter as "any new and useful art, machine, manufacture, or composition of matter, or any new or useful improvement [thereof]."

Act of Feb. 21, 1793, § 1, 1 Stat. 319.

The Act embodied Jefferson's philosophy that "ingenuity should receive a liberal encouragement." 5 Writings of Thomas Jefferson 75-76 (Washington ed. 1871). See *Graham v. John Deere Co.*, 383 U.S. 1, 7-10 (1966). Subsequent patent statutes in 1836, 1870, and 1874 employed this same broad language. In 1952, when the patent laws were recodified, Congress replaced the word "art" with "process," but otherwise left Jefferson's language intact. The Committee Reports accompanying the 1952 Act inform us that Congress intended statutory subject matter to "include anything under the sun that is made by man." S. REP. NO. 1979, 82d CONG. (1952); H.R. REP. NO. 1923, 82d CONG. (1952).

34. *Mortal Kombat* is a computer fighting game which was followed with a movie of the same name.

kits generally by themselves may be non-functional sub-components. *In re Venezia*³⁵ held patentable a "kit" having component parts capable of being assembled in the field.³⁶ In holding not-yet-cooperating kit elements patentable, this case, at least by analogy, indicates that a not-yet-cooperating computer component is also patentable. The court stated:

To hold that the words "any manufacture" exclude from their meaning groups or 'kits' of interrelated parts would have the practical effect of not only excluding from patent protection those 'kit' inventions which are capable of being claimed as a final assembly but also many inventions such as building blocks, construction sets, games, etc., which are incapable of being claimed as a final assembly.³⁷

Concededly, it is possible to claim software in the form of a "final assembly" (*i.e.*, a full computer system capable of executing the software); however, a computer readable medium programmed with computer instructions is undoubtedly a "building block" of a special purpose computer system. Moreover, claiming only the final assembly might undesirably limit direct infringers to end users.

The rationale of this kit case lends itself also to a "functional sub-component" theory which looks at software as a functional part of a machine. Ample authority supports the patentability of a single component of a machine as either a manufacture or as a machine itself: "[A] machine as a whole and also the sub-combinations that are contained therein are proper subjects of patents."³⁸ This theory, however, deals more with the "machine" aspect of software rather than the "manufacture" aspect. The "machine" approach to claims for computer instruction embodied in a computer readable medium is discussed again extensively in the following section. In dealing with the "manufacture" category, however, courts have applied an important doctrine relating to printed matter.

2. *The "Printed Matter" Doctrine*

Many thought that the printed matter doctrine was discarded and the patentability of "manufacture" claims to software was authoritatively resolved by *In re Beauregard*³⁹ (further discussed, *infra*). In reality, the Federal Circuit was precluded from making a definitive pronouncement on the issue. It appears, however, that manufacture claims to software will at least be allowed by the PTO since the Commissioner of Patents withdrew his rejections of Beauregard's claims in antic-

35. *In re Venezia*, 530 F.2d 956 (C.C.P.A. 1976).

36. *Id.* at 960.

37. *Id.*

38. 1 EARNEST B. LIPSCOMB III, WALKER ON PATENTS § 2.1 (3d ed. 1984) (citing *Cantrell v. Wallick*, 117 U.S. 689 (1886)).

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

ipation of the release of the *Guidelines* which permit such claims. The Commissioner's concession, however, forced the Federal Circuit to dismiss the case due to lack of actual controversy, leaving the issue unresolved by the courts.

Thus, prior Federal Circuit and Supreme Court cases are the only judicial pronouncements on the issue, and these fail to directly address functional code segments embodied in a computer readable medium. Nonetheless, much of the controversy regarding the printed matter doctrine and the type of claims which the PTO is willing to *issue* has now been allayed in view of *In re Warmerdam*,⁴⁰ *In re Lowry*,⁴¹ and particularly the *Guidelines*. An analysis of these decisions, however, demonstrates that the *Guidelines* take definite steps beyond the Federal Circuit's precedent. Thus, *In re Warmerdam*, *In re Lowry*, and other cases remain pertinent to the discussion of the validity and enforceability of certain claims to computer software embodied in a computer readable medium.

The term "computer readable medium" encompasses items such as compact disks, magnetic disks, magnetic tapes, and other similar articles which *by themselves* are statutory subject matter under § 101. Any remaining doubt about the patentability of such articles must arise from doctrines regarding whether their *content* may be the basis of their novelty. The judicially developed "printed matter" doctrine has long cast exactly that doubt upon the patentability of computer instruction embodied in a computer readable medium.

The early printed matter cases rejected items such as accounting forms.⁴² Any such printed matter on a piece of paper or other medium is generally disregarded and the underlying medium must be assessed for patentability. In most cases, the underlying medium (*e.g.*, paper, a compact disk, or a computer disk) is notoriously old and well beyond patentability. It is only in the cases where there is a "functional relationship of the printed material to the substrate" that the printed matter rejection is inapplicable.⁴³

One example where a printed matter rejection was reversed is *In re Gulack*.⁴⁴ In *In re Gulack*, the patentability of an "Educational and Recreational Mathematical Device in the Form of a Band, Ring or Concentric Rings" was at issue. Gulack's band could be used in a hat or other

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

41. *In re Lowry*, 32 F.3d 1579, 1580 (Fed. Cir. 1994).

42. *See, e.g.*, *United States Credit Sys. Co. v. American Credit Indem. Co.*, 59 F. 139 (2d Cir. 1893) (invalidating a patent drawn to sheets with headings adapted to preparing historical records of business transactions).

43. *In re Gulack*, 703 F.2d 1381, 1384 (Fed. Cir. 1983). *See also* *Cincinnati Traction Co. v. Pope*, 210 F. 443 (6th Cir. 1913).

44. *Gulack*, 703 F.2d at 1384.

article such as a wrist band. According to Gulack's specification, a band constructed in accordance with the appealed claims, "is capable of manipulation as set forth in the specification to perform magic tricks or to display various aspects of number theory."⁴⁵ Thus, the digits on the band had a functional relation with the band in permitting this "magic" display.

The claims were rejected under the printed matter doctrine because the device relied on numbers printed on the band for novelty.⁴⁶ The Federal Circuit reversed, stating that "[u]nder § 103, the board cannot dissect a claim, excise the printed matter from it, and declare the remaining portion of the mutilated claim to be unpatentable. The claim must be read as a whole."⁴⁷ Thus, the key to patentability in a printed matter case is whether the information cooperates or interacts functionally with the substrate on which it is imprinted or encoded, in a novel and non-obvious way.

In *In re Warmerdam*,⁴⁸ the Federal Circuit dealt with the question of whether a method of generating a hierarchy of bubbles is statutory under § 101. Claim 1 recited:

1. A method for generating a data structure which represents the shape of physical object in a position and/or motion control machine as a hierarchy of bubbles, comprising the steps of: first locating the medial axis of the object and then creating a hierarchy of bubbles on the medial axis.⁴⁹

Claims 2—4 recited both top-down and bottom-up procedures for creating the bubble hierarchy. The court determined that Claims 1—4 were non-statutory.⁵⁰ Notably, the Court seems to have tried to get away from the question of whether a mathematical algorithm was recited, stating "[t]he difficulty is that there is no clear agreement as to what is a 'mathematical algorithm' which makes rather dicey the determination of whether the claim as a whole is no more than that."⁵¹ The Court instead focused on whether the claimed method did any more than manipulate abstract ideas. It concluded that the claims did no more than call for such manipulation of abstract ideas and therefore, were not statutory subject matter.⁵²

Claim 5, on the other hand, recited:

45. *Id.* at 1383.

46. *Id.* at 1384.

47. *Id.* at 1385.

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

49. *Id.* at 1357.

50. *Id.* at 1360.

51. *Id.* at 1359.

52. *Id.* at 1360.

5. A *machine* having a memory which contains data representing a bubble hierarchy generated by the method of any of Claims 1 through 4.⁵³

As to Claim 5, the court stated that Claim 5 was drawn to a "machine" and was therefore, clearly patentable subject matter.⁵⁴ Thus, a machine holding a data structure was ruled patentable purely due to claim limitations relating to the data structure stored in memory. The memory by itself was in no way unique. Claim 5 was also challenged on definiteness grounds (*i.e.*, 35 U.S.C. § 112, paragraph 2); however, the Court reversed that rejection as well.⁵⁵ Thus, *In re Warmerdam* at least illustrates that a machine having a data structure in memory is patentable subject matter. Since the court considered the limitations on the structure in memory to overcome the § 112 rejections, this meant that the PTO was not free to ignore the data structures in the memory as the PTO previously attempted to do under the printed matter doctrine.

Shortly after its decision in *In re Warmerdam*, the Federal Circuit decided *In re Lowry*.⁵⁶ In *In re Lowry*, the Federal Circuit found that a memory storing data in a particular data structure is patentable, reversing a printed matter rejection.⁵⁷ The data stored by Lowry was included in a data structure which organizes the information in a database in accordance with an "attributive data model," which used attribute data objects ("ADOs") to organize the data.⁵⁸

According to Lowry's specification, an "[a]ttribute expresses the idea that one thing is attributed to another thing."⁵⁹ In such a model, information used by one or more application programs is organized in terms of attributes and the relationships of each attribute to the other attributes stored in the database.

Independent Claim 1 and dependent Claims 2—5 recite a memory for storing the data structure. Independent Claim 1 recites the following:

A memory for storing data for access by an application program being executed on a data processing system, comprising: a data structure stored in said memory, said data structure including information resident in a database used by said application program and including: a plurality of attribute data objects stored in said memory . . . ; a single holder attribute data object for each of said attribute data objects . . . ; a referent attribute data object for at least one of said attribute data objects . . . ; and an apex data object stored in said memory⁶⁰

53. *Id.* at 1358 (emphasis added).

54. *Id.* at 1360.

55. *Id.* at 1361.

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

57. *Id.* at 1584-85.

58. *Id.* at 1581.

59. *Id.* at 1580.

60. *Id.* at 1581.

The examiner rejected Claims 1—5 as being non-statutory under § 101 and as being obvious under § 103 in view of United States Patent No. 4,774,661 of Kumpati.⁶¹ The PTO Board reversed the § 101 rejection, concluding instead that the claimed memory storing the data structure was an article of manufacture.⁶² However, the PTO Board sustained the § 103 rejection relying on *In re Gulack* to disregard any recitation to the data structure.⁶³ The court summarized the rationale of the PTO Board as follows:

In *Gulack*, this court concluded that “the critical question is whether there exists any new and unobvious functional relationship between the printed matter and the substrate.” (citations and footnote omitted). The Board therefore framed the question as whether a new, nonobvious functional relationship exists between the printed matter (data structure with ADOs) and the substrate (memory). The Board determined that Lowry did not show such a functional relationship. Thus, the Board agreed with the examiner that the data structure could not distinguish the claimed invention from the prior art. The Board held that Kumpati, disclosing a CPU using a memory and containing stored data in a data structure, rendered all claims either anticipated or obvious. Lowry appealed.⁶⁴

The Federal Circuit, refusing this rationale, clarified that the printed matter cases were not relevant to the claimed data structures which purportedly required processing by a machine, stating:

The printed matter cases have no factual relevance where “the invention as defined by the claims requires that the information be processed not by the mind but by a machine, the computer.” (Citation omitted). Lowry’s data structures, which according to Lowry greatly facilitate data management by data processing systems, are processed by a machine. Indeed, they are not accessible other than through sophisticated software systems. The printed matter cases have no factual relevance here.⁶⁵

Nor are the data structures analogous to printed matter. Lowry’s ADOs do not represent merely underlying data in a database. ADOs contain both information used by application programs and information regarding their physical interrelationships within a memory. Lowry’s claims dictate how application programs manage information. Thus, Lowry’s claims define functional characteristics of the memory.⁶⁶

The court explained why Lowry did not merely claim information in memory:

61. *Id.* at 1582.

62. *Id.*

63. *Id.*

64. *Id.*

65. *Id.*

66. *Id.* at 1583 (emphasis added).

Contrary to the USPTO's assertion, Lowry does not claim merely the information content of a memory. Lowry's data structures, while including data resident in a database, depend only functionally on information content. While the information content affects the exact sequence of bits stored in accordance with Lowry's data structures, the claims require specific *electronic structural elements* which impart a *physical organization on the information stored in memory*. Lowry's invention manages information. As Lowry notes, the data structures provide increased computing efficiency.⁶⁷

Emphasizing the point, the court reiterated that "the data structures are specific electrical or magnetic *structural elements* in a memory" and as such "are *physical entities* that provide increased efficiency in computer operation."⁶⁸ These statements provide guidance for drafting claims to memories or other computer readable media containing data structures or, perhaps an application program. A claim reciting a memory storing "mere data" would not be allowed in view of the above comments, at least under a "printed matter" rejection, if not a § 101 rejection.

Thus, more than "mere data" is required for a claim to recite statutory subject matter. First, to avoid a "printed matter" rejection, a claim must "require" that the stored data be processed by a machine, *e.g.*, a computer. Second, there need not be a "relation between" the data and the memory as was required by the PTO Board in *In re Lowry*. Rather, the Federal Circuit requires a "physical organization on the data stored in memory." This physical organization is effected by the "electrical or magnetic structural elements" defined by the data structure's electronic form.

Unfortunately, the data structures being patented today can be quite complex, and the legal reasoning of cases such as *In re Lowry* may be lost in some of the technological detail. An example involving simpler data structures than that at issue in *In re Lowry* may be useful to better highlight the Federal Circuit's requirement that the electrical or magnetic structural elements impart a physical structure on the memory.

A simple array is a data structure which consists of a series of contiguous memory entries. As such, an array of a particular size consumes a known amount of contiguous memory entries. To access the Nth element of an array, one can index N times the element length into the memory and retrieve that item. Access is simplified by the contiguous nature of the array. While being contiguous improves array access ease, it complicates memory management.

67. *Id.* at 1583 (emphasis added).

68. *Id.* at 1583-84 (emphasis added).

A simple linked list is a data structure which remedies some of the shortcomings of a simple array. A linked list includes a series of data elements and pointers. Each data element includes data and each pointer either points to the next link or indicates that the data element is the last in the linked list. The linked list is advantageous because the pointer can point to a wide variety of memory locations which need not necessarily be contiguous. The end pointer can easily be updated, and new links are added to the linked list as necessary.

For each data element/pointer combination of the linked list stored in memory, there will be a first group of memory cells simply containing data (the data element) and a second group of memory cells containing the memory location of the next link. It is this second group of memory cells which provides the "electrical or magnetic structural elements" in the memory as described by the Federal Circuit. The ones and zeroes defining the memory location of the next link literally impose structure on the memory to the extent that when that memory location is read, this sequence of ones and zeroes (defining the location of the next link) will be returned. In other words, there is a structural change due to the electrical charges in the memory because this charge causes the memory to respond differently to input stimuli which read the memory.

This change in response, in and of itself, is not enough. Indeed, any sequence of ones and zeroes stored in a memory causes this type of change. For example, digitized music stored in a memory changes the response of the memory to input stimuli. It is additionally the memory organization inherent in the returned pointer values which differentiates the memory programmed with a linked list from a compact disk storing mere data. The values stored in memory impose an electrical structure which provides a physical organization of the memory that overcomes some of the limitations of an array. The linked list is a "physical organization on the information stored in memory" since each piece of data is accompanied by structural elements (i.e., electrical and magnetic charges) which point to a physically distinct location of the next piece of data. The Federal Circuit appears to have hit the nail on the head by pointing out that the electronic structure of the data structure physically organized the information stored in memory.

With this detailed understanding of the Federal Circuit's analysis applied to a simpler data structure, it is apparent that the *In re Lowry* decision was indeed well reasoned from a technical perspective. On the other hand, an understanding of what the Federal Circuit meant by "physical organization" and "electrical or magnetic structural elements" also leads to the conclusion that this rationale does not extend to computer instruction stored in memory (or any other computer readable medium).

The Federal Circuit *almost* had a chance to provide an authoritative statement on computer programs embodied in a computer readable medium in *In re Beauregard*.⁶⁹ Unfortunately, the Commissioner of the PTO effectively yanked jurisdiction out from under the Federal Circuit by withdrawing the standing rejection of the claims. The Federal Circuit accordingly dismissed the case since there no longer was any actual controversy.

Beauregard claimed a "computer program product" used in a computer system to perform the method of filling a polygon displayed on a graphics display device. Beauregard's claims are apparatus claims reciting an "article of manufacture," a "computer program product," or a "program storage device." Although these claims and the specification make it clear that a computer would execute the computer program, no computer is explicitly recited in the claims.

The PTO Board of Patent Appeals and Interferences affirmed the Examiner's final rejection of all pending claims. The PTO Board agreed with the Examiner that the computer product claims were "printed matter." Accordingly, the claims were rejected as being nonstatutory under § 101 and obvious under § 103.⁷⁰

Embracing the standard printed matter rationale, the PTO Board considered the program code to be analogous to the text of a book, and the computer readable medium to be analogous to the paper which contained the text. By applying this analysis, the PTO Board concluded that Beauregard claimed "text on paper" which was unpatentable "printed matter."⁷¹ The PTO Board reached this conclusion despite the fact that the Beauregard claims expressly provided that the program code would "cause" the "computer to effect" certain functions in the computer. By focusing on the "printed matter" aspect of the invention, the PTO Board failed to consider the claims as a whole.

The *In re Lowry* decision came after the PTO Board's decision on Beauregard's claims, but before the claims reached the Federal Circuit. Thus, the decision in *In re Lowry* foreshadowed a reversal of the PTO Board's application of the printed matter rejection in the *In re Beauregard* case, since the Federal Circuit reversed a printed matter rejection in *Lowry*.

Applying the detailed rationale of *In re Lowry* to Beauregard's claims, however, the result may not be so clear. In *In re Lowry*, the Federal Circuit held that "printed matter" rejections do not apply to "information stored in memory" if "the invention as defined by the claims requires that the information be processed not by the mind but by a

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

70. *Ex Parte Beauregard*, unpublished opinion No. 93-0378 (July 29, 1993).

71. *Id.* at 6.

machine, the computer.”⁷² However, the Federal Circuit in *In re Lowry* also stated that the claims “require specific electronic structural elements which impart a physical organization on the information stored in memory.”⁷³

Software, as stored in a memory, constitutes a sequence of ones and zeroes which are interpreted by a processor to be instructions. As in the case of a data structure, the storage of the software in the memory causes the memory structure to become altered by the “electrical or magnetic” representation of the instructions stored therein. Both may be said to affect the way the computer functions. The similarities cease there. A data structure is an item in memory which perhaps improves the utilization of the memory. Being stored in memory is its end, and a computer executing a program to create the data structure is the means. The computer program, on the other hand, is only stored in memory *in order to be executed*. Execution by the processor is the ends of a computer program and storage in memory is only a necessary precondition.

The instructions stored in memory do not directly reflect relationships between different memory locations. In other words, strictly speaking, software instruction does not enhance the utility of the memory by reconfiguring the memory, as a data structure does. Rather the software instructions *reconfigure the processor* to perform specific tasks. Although the memory is reconfigured with the ones and zeroes, this in and of itself does not create a “physical organization of the information stored in the memory” as did the data structure in *In re Lowry*.

One might argue that the difference is merely hair splitting. Indeed, the *In re Lowry* decision invokes some broad language. For example, the court did state that “Lowry’s claims define functional characteristics of the memory.”⁷⁴ In a broad sense, a computer program may be considered to define functional characteristics of the memory. However, a computer program may more accurately define the functional characteristics of the entire computer system or of the processor which executes the program.

Thus, after a detailed analysis of *In re Lowry*, and a technical analysis of the differences between data structures and programs, it is apparent that there is not a definitive answer to the question of the patentability of software embodied in a computer readable medium. Moreover, it appears questionable as to whether the *In re Lowry* rationale really applies.

72. *In re Lowry*, 32 F.3d 1579, 1583 (Fed. Cir. 1994) (emphasis added).

73. *Id.*

74. *Id.*

B. SOFTWARE AS A SUBSTITUTE FOR PHYSICAL STRUCTURE

As discussed above, computer instruction differs from a data structure in a computer memory in a rather profound manner. The differences make the detailed rationale provided with respect to data structures in *In re Lowry* not fully applicable to computer instruction. Nonetheless, computer instruction embodied in a computer readable medium does qualify as statutory subject matter under § 101, simply because the storage of the computer instruction turns a computer readable medium into a *functional component* which directly cooperates with the processor. The following discussion describes why software embodied in a computer readable medium is actually a functional component, and accordingly, deserves patent protection. This rationale is more appropriate for computer instruction than the *In re Lowry* rationale which applied to a data structure.

The fact that a computer readable medium is no less a physical structure with functional characteristics than other computer components, is illustrated by understanding the role computer programs played in the development of general purpose computers. Without engaging in detailed legal analysis for a moment, one can easily appreciate that the first transistor was clearly statutory subject matter.⁷⁵ Indeed, this simple device functioned as an electrically controlled switch which allowed a current to flow between two electrodes when a certain electrical potential was applied to a third. The ensuing years brought simple logic gates. Such gates combined two or more input values to form output values based on these inputs. For example, a two input AND gate is a binary circuit that generates an output value one if both of its two inputs receive a one. The first AND gate was also no doubt patentable subject matter.

Armed with logic circuits, designers created circuits known as combinational logic circuits which would repeatedly respond to input stimuli in a known manner. For example, an adder circuit added two values. Importantly, it was possible to either choose the values to be added in advance by wiring the inputs to the circuit to logical ones and zeroes, or to input values during operation. Undoubtedly, the first such adding circuit was also patentable as a machine.

As the ability to store ones and zeroes developed (*i.e.*, memory circuits), combinational logic circuits evolved into sequential logic circuits—circuits with the ability to perform a sequence of operations based on stored values and a clocking mechanism. In some cases, circuits which performed specific tasks were designed; however, as the complexity and

75. See U.S. Patent No. 2,981,877, "Semiconductor Device—and—Lead Structure" issued to Robert Noyce in 1961.

variety of such tasks grew, designing and fabricating the multiplicity of special purpose circuits that were needed became prohibitive.

Thus, the need for general purpose processing devices arose. Devices changed from being entirely "hard wired," meaning that all decisions were built into the physical circuitry, to being programmable, meaning that a variety of functions depended on user input. Although special purpose circuits could have been designed for all of the particular purposes, it became easier to mass produce general purpose circuits that were configured by user inputs (*i.e.*, machine instructions) to perform the same function as the special purpose circuit. Machine instructions configure the machine to perform a special purpose.⁷⁶

Designers of course realized that the value of both combinational and sequential logic circuits was that the same circuit could process a multitude of different input stimuli. Soon, a general purpose processor developed which could move values around in memory, perform arithmetic operations on them, and transfer them to various other devices for display or other type of output. As an example, the processors contained circuits to perform addition.⁷⁷ In the early days, if a computer operator desired to perform a complicated mathematical operation, the operation would have to be broken down into simple operations which the available hardware was capable of performing. For example, if a programmer wished to compute the tangent of an angle, a series of multiplies, additions, and subtractions may have been necessary to perform a mathematical approximation of the tangent function.

As designs grew in complexity, more and more functions were integrated into math coprocessors,⁷⁸ and later into microprocessors.⁷⁹ For many years, there have been great debates about whether it is preferable to integrate many functions into a complex microprocessor or to rely on the speed of a simple and faster microprocessor. The former school of thought, the complex instruction set computing ("CISC") school, provides programmers with a wide array of commands to perform many tasks. The result is short programs but more complicated hardware. Reduced instruction set computing ("RISC"), on the other hand, generally pro-

76. Indeed the courts have recognized that programming a general purpose computer with a specific program configures that computer into a special purpose computer for performing the program function. *See, e.g., In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994).

77. For example, the first microprocessor, the Intel 4004 central processing unit, included an on board addition circuit ("ALU") capable of adding four bit numbers. *See, e.g., KENNETH L. SHORT, MICROPROCESSORS AND PROGRAMMED LOGIC 5* (2d ed. 1987).

78. The Intel 8087 is an example of one of the first popular math coprocessor circuits which used approximation algorithms to compute transcendental functions such as tangent. *See, e.g., V. CARL HAMACHER ET AL., COMPUTER ORGANIZATION 304* (3d ed. 1990).

79. *Id.* at 423-24. The 80486 and subsequent generations of Intel microprocessors integrated a floating point unit capable of such transcendental function computations.

vides programmers with a smaller set of commands, thus requiring longer code segments to achieve the desired ends. The RISC school relies on the higher processing speeds of simpler hardware to compensate for such lengthy programs.

The relevance of this long-standing debate is that computer architects have long recognized that computer hardware and computer software are directly interchangeable. A RISC machine has simpler processing hardware and longer software programs when compared to a CISC machine performing the same function. The components necessary for a CISC machine that performs a transcendental function may simply be the computer system and one single instruction. In a RISC machine, the components may include a sequence of code of some length and a simpler processor in the computer system. The transcendental function computation routine in the RISC machine is no less a component of the system than is the circuit in the CISC machine. The difference is only the reflection of a design preference.

Thus, from a technical perspective, a computer intended to accomplish a particular purpose could conceivably be designed solely in what is commonly known as hardware, or as a combination of hardware and software. The evolution and mass production of computer components and products dictates the use of general purpose hardware to produce commercially viable products. As a result, components of the complete machine reside on magnetic or other computer readable media. Such components are no less integral parts of the total machine than are the visible hardware components. For the lay person, the relative ease of changing software (compared to hardware) masks the profound truth that you are reconfiguring your machine to *function* in a different way when you do so. Additional components (new software) may be added to improve functionality in the same manner as adding new peripheral devices (e.g., a color printer).

Thus, software forms an integral part of the functional engine of each general purpose computer. Just as an integral part of a more physically palpable machine is patentable, so is the software component of each special purpose computer, so long as it is physically embodied in some medium. As far as placing software in one of the categories of § 101, a computer readable medium with embedded functional software may be viewed as an article of manufacture, a machine, or both. The "proper" categorization, if there is one, is admittedly a bit vexing, as a disk with software on it appears to fall within both the generally accepted definitions of manufacture and machine (*i.e.*, as a sub-component of a machine).

Fortunately, the categorization is largely irrelevant since both categories are equally entitled to patent protection under § 101. Perhaps the PTO has found the best solution. The *Guidelines* generally do not distin-

guish between "manufacture" and "machine," but rather refer to claims implicating *either* category as "product" claims. So call it a manufacture, or call it a machine, or just admit that it is something "useful" and "under the sun." Software embodied in a computer readable medium is patentable subject matter.

C. PATENTABILITY OF COMPUTER READABLE MEDIA UNDER THE PTO GUIDELINES

After a painful fight to establish software embodied in a computer readable medium as statutory subject matter, one must be cautious to avoid tripping over other hurdles to gaining patent protection for computer related inventions. The two important issues under the *Guidelines* with respect to software embodied in a computer readable medium are (1) the evaluation of the underlying process; and (2) the extent to which the *Guidelines* extend beyond the holding in *In re Lowry*.

As to the first issue, the *Guidelines* generally follow judicial precedent closely regarding process evaluation set forth above. The *Guidelines* consider the underlying process and require that the Examiner determine what is claimed.⁸⁰ The claimed invention may not be a "natural phenomenon" as mandated by *Diehr*.⁸¹ Additionally, the underlying process in such claims must not "merely manipulate an abstract idea or solve a purely mathematical problem without any limitation to a practical application."⁸² Rather, the process must perform "independent physical acts" such as "post-computer process activity" or must "manipulate data representing physical objects" or include "activities to achieve a practical application" such as "pre-computer process activity."⁸³ Thus, as to process evaluation, the *Guidelines* very closely track binding precedent on the issue.

As to the second issue, whether they extend beyond the holding in *In re Lowry*, the *Guidelines* appear to do so, and a practitioner may be well advised to adhere closely to judicially validated claim constructs. The *Guidelines* discuss articles of manufacture containing "descriptive material."⁸⁴ In accordance with *Diehr*, the examiner is directed to consider any claimed descriptive material.⁸⁵ First, the examiner is to classify it as either functional or non-functional.⁸⁶ Either data structures or computer instruction would appear to fall within the "functional" categoriza-

80. *Guidelines*, *supra* note 1, at 99.

81. *Guidelines*, *supra* note 1, at 100.

82. *Guidelines*, *supra* note 1, at 100.

83. *Guidelines*, *supra* note 1, at 100.

84. *Guidelines*, *supra* note 1, at 89.

85. *Guidelines*, *supra* note 1, at 89.

86. *Guidelines*, *supra* note 1, at 89.

tion as they both have functional effects on a computer system and/or the memory.

The *Guidelines* state, "[b]oth types of 'descriptive material' are non-statutory when claimed as descriptive material per se."⁸⁷ Even if the "descriptive material" is "embodied in computer-readable media," only embodied "functional descriptive material" is statutory.⁸⁸ The *Guidelines* provide:

When functional descriptive material is recorded on some computer-readable medium it becomes *structurally and functionally* interrelated to the medium and will be statutory in most cases. When non-functional descriptive material is recorded on some computer-readable medium, it is not structurally and functionally interrelated to the medium but is merely carried by the medium. Merely claiming non-functional descriptive material stored in a computer readable medium does not make it statutory. Such a result would exalt form over substance. Thus, non-statutory music does not become statutory by merely recording it on a compact disk. Protection for this type of work is provided under the copyright law.⁸⁹

It is clear that the *Guidelines'* conclusion extends beyond that of *In re Lowry*. The *In re Lowry* decision was based on the functional relationships of the electronic structural elements whose arrangement imparted a physical organization on the memory. A data structure qualifies as structurally interrelated with the medium. It is not clear that a data structure is really functionally interrelated to the medium, since the data structure itself does not cause computer functions to occur. Computer instructions, on the other hand, do cause computer functions to occur and are therefore, more akin to a functional component of the computer system. This makes computer instruction functionally interrelated with the medium in the sense that the computer readable medium becomes a functional component of the computer system by way of the software embedded thereon.

Thus, the *Guidelines'* conjunctive recitation of "structurally *and* functionally" is troublesome because data structures may not be functionally interrelated with the medium. Furthermore, the instructions stored in memory, in the case of a program, do not impose a defined structure on the memory (as does a data structure), but rather impose functional characteristics on the computer system.

Even though a disjunctive "structurally *or* functionally" arguably is more appropriate, the *Guidelines'* language *is* defensible. As to data structures, arguably a data structure can change the overall function of the computer system when coupled with the necessary software. Indeed,

87. *Guidelines*, *supra* note 1, at 89.

88. *Guidelines*, *supra* note 1, at 89.

89. *Guidelines*, *supra* note 1, at 89 (emphasis added).

In re Lowry discussed the improved manner in which a system using Lowry's data structure functioned. As to the instructions, since the computer instruction is structurally interrelated to the medium by way of its storage thereon, and by the inherent program structure (i.e., loops, branches, instruction sequence), the language is arguably justified. Thus, the *Guidelines* offer technically and legally sound justifications for patenting data structures and software embodied in a computer readable medium. However, these justifications have not passed through the judicial wringer.

The *Guidelines'* conclusions can be further tested by applying an interesting hypothetical extreme. Consider a future time when a computer is fully capable of reading and interpreting handwriting. Will a programmer's handwritten scrawl of a computer program, or even a flowchart on a napkin, become a patentable item because the napkin has become a computer readable medium with computer instruction embodied thereon? This case beckons for application of the abandoned "printed matter" rationale to disregard the print and find the napkin non-novel.

Nonetheless, under the *Guidelines*, the handwritten computer instruction *will* qualify as "functional descriptive matter" once computer systems can automatically scan the napkin, decipher the handwriting, and compile and execute the code. In the not-so-distant future, this process may be as easy as it is now to put a CD-ROM into the proper port and execute a program thereon. At that point, a napkin with a handwritten program would appear to be no less patentable under the *Guidelines* than a storing a program.

D. CHOOSING APPROPRIATE CLAIM LANGUAGE

In view of the questionable result of a patentable napkin, the practitioner may be well advised to choose slightly more restrictive claim language, at least until broad constructs such as claims to "computer readable media" are judicially validated. Choosing judicially validated claim language is likely the best prophylactic in view of the remaining uncertainty regarding such claims' validity. Fortunately, a practitioner can take steps to improve the likelihood that claims will be upheld without sacrificing significant claim breadth.

1. *The "Execution" Requirement*

Under *In re Lowry*, the printed matter doctrine did not apply where the claims required processing of information by a computer. If indeed our napkin requires processing by a computer, the *Guidelines'* result may be correct. The problem is that it is virtually impossible to draft meaningful claims that require execution of instructions by a computer without claiming the computer itself. Consider the following claim:

An article of manufacture with a computer program embodied therein, the computer program comprising: a first code segment for sorting a plurality of entries; and a second code segment for displaying the plurality of entries. This claim clearly does not require a computer, but appears to be patentable subject matter under the *Guidelines*.

Now consider also Beauregard's Claim 10: a program storage device readable by a machine, tangibly embodying a program of instructions *executable* by the machine to perform method steps . . . comprising: testing the polygon . . .; * * * combining . . .; and passing a pointer . . .⁹⁰

Also, consider Lowry's allowed Claim 1, which recites: a memory for storing data *for access* by an application program *being executed* on a data processing system, comprising: "a data structure stored in said memory, said data structure including information resident in a database used by said application program and including a plurality of attribute data objects."⁹¹

Does the mere fact that Beauregard's program of instructions is executable by a machine mean that the claim "requires" such execution? Apparently, Lowry's memory "for access" by "an application program being executed on a data processing system" is enough.

Although, arguably, Beauregard's Claim 10 could be read on a napkin at some point in the future, the author believes the right line has been drawn. Perhaps the courts will need to breathe life back into the printed matter doctrine, to the extent that a patentee attempts to enforce software claims against an item that could possibly be deemed to carry "functional descriptive matter," if in fact the item is not intended for use as a functional item, as in the case of the textbook.

Alternatively, the courts could also disregard cases where the claimed use is in fact commercially impracticable. For example, it is exceedingly unlikely that a textbook would be marketed as a vehicle for selling an operative computer program when a disk or even an Internet transmission is infinitely more practical for that purpose. Unfortunately, any such analysis grafted upon a direct infringement analysis reduces the simplicity of enforcement which makes computer readable medium claims attractive in the first place.

The alternative of literally requiring execution also seems untenable if claims to computer programs embodied in a computer readable medium are to be of any use whatsoever. First, if the claims explicitly require that execution take place, the only infringers will be end users who run the software. This forces the patentee to rely on theories of contributory or induced infringement for enforcement against software develop-

90. U.S. Patent No. 5,710,578, Claim 10 (emphasis added).

91. U.S. Patent No. 5,664,177, Claim 1 (emphasis added).

ers or distributors who are more likely to be the patentee's primary target.

Second, if claims absolutely require that the computer readable medium can only be readable by a machine executing the program, an infringer could create a device that allows the computer instruction to be read and displayed, but not executed. With such a device, the alleged infringer can argue that his software embodied in a computer readable medium is both executable and separately readable and displayable without execution, thus not infringing upon a claim for a program which can only be executed and not otherwise read.

Thus, the following claim may be criticized on several grounds. An article of manufacture with a computer program embodied therein, the computer program comprising a first code segment for sorting a plurality of entries and a second code segment for displaying the plurality of entries.

First, it does not require, or even suggest, execution by a machine. Since the execution "requirement" in *In re Lowry* appears, in reality, to only require a suggestion of machine execution, the inclusion of such a suggestion is recommended. Alternatively, the proponent of such a claim may rely on the sub-component of a machine theory elucidated above. However, this leads to the second criticism. The recitation of "article of manufacture" to some extent undermines the argument that the claim is drawn to a machine or a sub-component thereof.

A better solution is the following: a machine readable medium having embodied thereon a computer program for processing by a machine, the computer program comprising a first code segment for sorting a plurality of entries and a second code segment for displaying the plurality of entries.

This claim requires (at least as much as *In re Lowry* did) machine activity. The machine activity is simply referred to as "processing," which would include an infringer's use of any machine to read or execute the data. Moreover, a machine-readable medium is recited, leaving ambiguous whether the applicant considers the claimed item to be a manufacture or a machine. To more closely track the language of *Lowry* and *Beauregard's* claims, "for execution" may be used instead of "for processing." The fact that the computer readable medium contains a computer program "for execution" does not preclude other uses, such as reading or reverse engineering the computer program, as it is not claimed "only for execution."

Finally, it may be desirable to include the computer readable medium in the text of the claim *after* the transition word (*e.g.*, comprising), since courts may look at the preamble as a field of use restriction or otherwise as an improper basis for the sole structure which makes a

claim statutory. Indeed, if the computer readable medium does not follow the transition word, the body of the claim contains only disembodied software. Unfortunately, including the computer readable medium requires an additional noun before the transition word. An "embodiment of a program" is recommended, although "article" may be used.

2. *The "Steps for" Quagmire*

An additional popular way to claim software embodied in a computer readable medium is to recite program functions using a method-like construct. Unfortunately, this format may be unduly limiting under a recent Federal Circuit decision, and thus should be avoided or at least should be used cautiously. An example of a claim arguably in the "step plus function" format is Claim 10 of *In re Beauregard*,⁹² or the following: a computer readable medium having a computer program embodied thereon, the computer program being executable by a machine to perform the steps of sorting a plurality of entries by invoking a routine to sequence a plurality of numeric equivalents for said plurality of entries and displaying the plurality of entries.

In *O.I. Corp. v. Tekmar Co., Inc.*,⁹³ the Federal Circuit recently interpreted § 112, paragraph 6, with respect to method claims reciting "steps" for performing a particular function. Indeed, paragraph 6 provides:

An element in a claim for a combination may be expressed as a means or *step* for performing a specified function without the recital of structure, material, or *acts* in support thereof, and such claim shall be construed to cover the corresponding structure, material, or *acts* described in the specification and equivalents thereof.⁹⁴

The Federal Circuit stated, "[w]e interpret the term 'steps' to refer to the generic description of elements of a process, and the term 'acts' to refer to the implementation of such steps."⁹⁵ Thus, the Federal Circuit linked "acts" to steps much like "means" is linked to "structure" and "material."

While it is well established that the applicant using broad and easily drafted "means for" claims bears the price of being limited to the structure in the specification under § 112, Paragraph 6,⁹⁶ the Federal Circuit clarified that "[s]imilarly, a step for accomplishing a particular function

92. U.S. Patent No. 5,710,578, Claim 10 (emphasis added). Notably, some claim elements of the Beauregard patent are compound elements that include what may be deemed a "function" in addition to the initial gerund. *Id.* For example, "testing the polygon to determine if there is one continuous scan line for each one of a plurality of scan lines of said polygon" may arguably include the "to determine" function.

93. *O.I. Corp. v. Tekmar Co., Inc.*, 115 F.3d 1576 (Fed. Cir. 1997).

94. 35 U.S.C. § 112 (1994) (emphasis added).

95. *Tekmar*, 115 F.3d at 1582.

96. *See, e.g., In re Donaldson*, 16 F.3d 1189 (Fed. Cir. 1994).

in a process claim may also be claimed without specificity subject to the same price."⁹⁷

Thus, it is possible to construe "steps for" functions as being limited to the disclosed "acts" in the specification. It does appear, however, that simple process steps are "acts" rather than functions, and that to invoke Paragraph 6, a complex step element reciting a function beyond the action directly recited by the initial gerund is required.⁹⁸ For example, the above claim for sorting and displaying could be interpreted under Paragraph 6 if "invoking a routine to sequence a plurality of numeric equivalents" or "sequenc[ing] a plurality of numeric equivalents" is a "function" and not an "act."

Since software claims are often drafted with complex method step elements, applicants may be surprised by the limiting interpretation many claims may ultimately receive. For example, if the applicant details the exact acts to invoke the routine or perform the sequencing in a particular bubble sort routine, the claim may be limited to that sorting technique. Worse yet, if the applicant includes the source code of the program, the claim may be construed even more narrowly to cover the precise sequence of instructions in the code.

It remains far from clear when a particular element in a claim is properly interpreted as a "step plus function." A practitioner is likely left guessing how a particular court will apply Paragraph 6. Although it appears there is no magic language that definitively invokes Paragraph 6,⁹⁹ it does appear that the words "means" and "steps" do have some mystical and perhaps dangerous qualities. Certainly, the use of such language invites interpretation under Paragraph 6.

Thus, it behooves the practitioner attempting to obtain the broadest possible claims to minimize the risk of a narrowing application of Paragraph 6. This must be done by avoiding the "steps for" construct in claims. Considering Lowry's machine-processing prong, and the dangers

97. *Tekmar*, 115 F.3d at 1583.

98. *See Serrano v. Telular*, 111 F.3d 1578, 1583 (Fed. Cir. 1997). The court construed a "determining" step as "the only act" while a similarly worded "determination means" was construed under paragraph 6. *Id.* at 1580. The determination-means and determination-step read: "determination-means coupled with the telephone number digital conversion means for automatically determining the last digit of the group of telephone digits provided at the transceiver coupling means . . . automatically determining at least the last-dialed number of the telephone number dialed on the telephone communications-type device." *Id.*

99. *Cole v. Kimberly-Clark Corp.*, 102 F.3d 524, 531 (Fed. Cir. 1996). The court stated: [m]erely because a named element of a patent claim is followed by the word "means," however, does not automatically make that element a "means-plus-function" element under 35 U.S.C. § 112," and that "[t]he converse is also true; merely because an element does not include the word "means" does not automatically prevent that element from being construed as a means-plus-function element.

Id.

of "steps for" language under Paragraph 6 of § 112, a claim to software embodied in an article of manufacture should be claimed, in substance, in one of the following two manners:

1. A machine readable medium having embodied thereon a program for execution by a machine, the program comprising: a first code segment for sorting a plurality of entries; and a second code segment for displaying the plurality of entries.

2. An article¹⁰⁰ comprising a machine readable medium having embodied thereon the program, the program being executable by a machine to perform: sorting a plurality of entries; and displaying the plurality of entries.

Neither claim recites "steps" and both require machine execution.¹⁰¹ The latter is preferred, as the machine-readable medium follows the transition word, and the elimination of the code "segments" concept precludes subsequent controversies regarding whether particular functions are encompassed in particular segments.

Furthermore, practitioners should thoughtfully consider the level of detail disclosed in the specification in view of the uncertainty regarding the application of § 112, Paragraph 6, to method claims. In particular, it may be wise to *reduce* the amount of disclosure so that there are no particular acts in the specification. For the example involving a sorting algorithm, it may be prudent to state only that "any sorting algorithm may be used." This language should be sufficient to satisfy the enablement and best mode requirements, since one of ordinary skill in the computer programming field is typically attributed with sufficient skills to implement a program from a high level description. While disclosure of only this high level description appears contrary to the policy of disclosure evidenced by the Constitutional basis for the patent law, "Promoting the Progress of Science,"¹⁰² it nonetheless appears to be the present reality with respect to § 112, Paragraph 6.

One final note regarding claim breadth is that the practitioner should certainly avail himself or herself of the PTO's broad view of what constitutes a computer readable medium. In particular, instruction materials developed in conjunction with the *Guidelines* suggest that

100. An "an embodiment of a program" may be used instead of "article" because, as discussed previously, using the term "article" undermines the argument that the invention is patentable because it is a functional component of a *machine*, since machines are a class of statutory subject matter enumerated separately from articles in 35 U.S.C. § 101.

101. Concededly, the mere elimination of the words "steps for" should not change the interpretation of a claim. The majority opinion in *Kimberly-Clark* admonished that using particular means language does not create a presumption as posited by Judge Rader's dissent. *Kimberly-Clark*, 102 F.3d at 531. Nonetheless, the language recited in the statute certainly at least draws attention to the issue.

102. U.S. CONST. art. I, § 8, cl. 8.

computer instruction embodied in a "carrier wave" is embodied in a computer readable medium,¹⁰³ and at least one patent claiming a signal embodied in a memory has been issued.¹⁰⁴ Since Internet transmissions are becoming an increasingly important channel of software distribution,¹⁰⁵ it is highly recommended that either (1) the computer readable medium being a carrier wave be recited at least in a dependent claim; or (2) the specification include a very clear and very broad definition of computer readable medium, making it clear that one who sells a transmission over the Internet will be a direct infringer of claims to the software embodied in a computer readable medium.

E. THE APPLICATION OF OTHER STATUTORY REQUIREMENTS FOR SOFTWARE CLAIMS

A cautious practitioner should continually bear in mind that the *Guidelines* do not have the force of law, and (if followed) may only result in easier *issuance* of software patents. Defendants in infringement suits, however, are free to litigate all validity issues. Having addressed the *Guidelines'* vulnerability where the limits of case law are exceeded, it is also prudent to examine the implications of § 112 as they specifically apply to patenting software. The good news is that case law has addressed a number of these issues, and § 112 does not pose as substantial a barrier to patenting software as § 101 once did.

1. *The Enablement Requirement*

The enablement requirement derives from the first paragraph of § 112 which provides:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to *enable* any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor of carrying out his invention.¹⁰⁶

A decision on the issue of enablement requires a determination of whether a person skilled in the pertinent art, using the knowledge available to such a person and the disclosure in the patent document, could make and use the invention without undue experimentation. It is not

103. PAT. TRADEMARK OFF., example claim 13; AUTOMATED MANUFACTURING PLANT, Mar. 28, 1996, at 37.

104. See U.S. Patent No. 5,568,202, "System for Echo Cancellation Comprising an Improved Ghost Cancellation Reference Signal."

105. For example, on January 28, 1998, Egghead software announced its intent to close all retail outlets and solely rely on Internet distribution. See, e.g., *Egghead To Close Stores, Expand Sales On Web*, L. A. TIMES, Jan. 29, 1998, at 1.

106. 35 USC § 112 (1994) (emphasis added).

fatal if some experimentation is needed, for the patent document is not intended to be a production specification.¹⁰⁷

Fortunately for patentees, the courts have been rather generous in assuming that a skilled computer programmer can derive a functional program from a high level description. However, this has not always been true, and each case will turn on its own facts. The Federal Circuit has stated that "[t]he amount of disclosure that will enable practice of an invention that utilizes a computer program may vary according to the nature of the invention, the role of the program in carrying it out, and the complexity of the contemplated programming, all from the viewpoint of the skilled programmer."¹⁰⁸

Most favorable, however, is the language in *In re Sherwood*,¹⁰⁹ in which the Court stated:

In assessing any computer-related invention, it must be remembered that the programming is done in a computer language. The computer language is not a conjuration of some black art, it is simply a highly structured language The conversion of a complete thought (as expressed in English and mathematics, i.e. the known input, the desired output, the mathematical expressions needed and the methods of using those expressions) into a language a machine understands is necessarily a mere clerical function to a skilled programmer.¹¹⁰

Since reducing a flowchart, or a generally known concept such as a "bubble sort," has been deemed a "mere clerical function to a skilled programmer," it would appear that only for inventions where significant inventive aspects relate to the exact coding used, would disclosure of such coding be required to satisfy the enablement requirement. In view of the drawbacks of such detailed disclosure described above, it would generally seem advisable to avoid disclosing source code or other such detailed information.

There have, however, been circumstances where creation of a computer program was not routine. For example, in *White Consolidated Industries, Inc. v. Vega Servo-Control, Inc.*,¹¹¹ the Court found that where a disclosure left one and one half to two person years of work to implement the appropriate program, the enablement requirement had not

107. *Northern Telecom, Inc. v. Datapoint Corp.*, 908 F.2d 931, 941 (Fed. Cir. 1990) (citing *In re Gay*, 309 F.2d 769, 774 (C.C.P.A. 1962)). See also *Atlas Powder Co. v. E.I. Du Pont De Nemours & Co.*, 750 F.2d 1569, 1576 (Fed. Cir. 1984) (discussing the required amount of experimentation).

108. *Northern Telecom*, 908 F.2d at 941 (citing *In re Sherwood*, 613 F.2d 809, 817 (C.C.P.A. 1980)).

109. *In re Sherwood*, 613 F.2d 809, 817 (C.C.P.A. 1980).

110. *Id.* at n. 6 (emphasis added).

111. *White Consol. Indus., Inc. v. Vega Servo-Control, Inc.*, 713 F.2d 788 (Fed. Cir. 1983).

been met.¹¹² This case, however, appears to be the exception rather than the rule, and the enablement requirement generally poses no more, and perhaps less, of a substantial hurdle with respect to software related inventions than other types of subject matter.

2. *The Description Requirement*

The description requirement, also derived from § 112,¹¹³ does not pose any particularly unique threats to patent claims which are drawn to software embodied in a computer readable medium. The description requirement generally enforces the policy that the public should be put on notice of what the applicant is entitled to claim according to what is *described* in his *original* disclosure. “[T]he applicant must . . . convey with reasonable clarity to those skilled in the art that, as of the filing date sought, he or she was in possession of the invention.”¹¹⁴

This requirement usually bears its teeth where the applicant seeks to claim subject matter that was not claimed when the application was filed. If so, the patent may be invalidated if those of skill in the art would not have been on notice that the applicant might claim the claimed subject matter based on the filed disclosure.

The description requirement, of course, is good motivation for an applicant to disclose as much as possible since claims may need to be narrowed or otherwise altered during prosecution; however, the benefits of increasing disclosure must be weighed against the potential limiting effects of disclosing too many details. The practitioner should carefully evaluate all possible claim scopes that are worth pursuing for the client. There should be given enough support for such claims, and perhaps no more.

3. *Definiteness under 35 U.S.C. § 112, Second Paragraph*

Under the second paragraph of 35 U.S.C. § 112, claims are required to “particularly point out and distinctly claim the subject matter which the applicant regards as his invention.”¹¹⁵ The purpose of this requirement is to put the public on notice as to what constitutes infringement of the patent.¹¹⁶ Additionally, this requirement “provide[s] a clear measure of the invention in order to facilitate determinations of

112. *Id.* at 791.

113. Section 112 provides “The specification shall contain a written description of the invention . . .” (emphasis added). 35 U.S.C. § 112 (1994).

114. *Vas Cath, Inc. v. Mahurkar*, 935 F.2d 1555, 1563-64 (Fed. Cir. 1991).

115. 35 U.S.C. § 112.

116. *United Carbon Co. v. Binney Co.*, 317 U.S. 228 (1942) (“A zone of uncertainty which enterprise and experimentation may enter only at the risk of infringement [sic] claims would discourage invention only a little less than unequivocal foreclosure of the field.”).

patentability."¹¹⁷

Since the description of a software program "product" really amounts to a description of the steps performed by the program, claims drafted to describe such a product by reciting the program steps can satisfy the definiteness requirement of § 112. While no Federal Circuit case directly discusses the definiteness requirement with respect to computer instructions, *In re Warmerdam*¹¹⁸ does discuss paragraph 2 in the context of data structures.

Warmerdam's claims did not describe any technique for configuring the memory, but rather recited the data structure itself. The court found that a claim description of *just the data structure* is sufficient to allow one of skill in the art to determine if a memory containing the data structure is covered by the claim.¹¹⁹ Thus, the definiteness requirement of § 112 can be met by simply reciting the elements of a data structure in terms convenient to and understandable by those in the art. In *In re Warmerdam*, the convenient mechanism for describing the data structure was the method by which it was produced.

This rationale appropriately extends to computer programs. A description of a computer program by the steps the program performs are certainly understandable by one in the field of computer programming. A skilled programmer writing a program undoubtedly could determine whether a computer disk containing a known program would be within the confines of a claim that listed a series of computer program steps. Thus, an applicant is free to claim software embodied in a computer readable medium in its natural state. That is, the applicant may claim the software by enumerating the steps that it performs.

4. *The Best Mode Requirement*

Finally, § 112 imposes a best mode requirement. In particular, § 112, first paragraph, requires that the specification "set forth the best mode contemplated by the inventor of carrying out his invention."¹²⁰ The best mode requirement is part of the quid pro quo of obtaining a monopoly for one's invention.¹²¹ The public must be informed of the best way to accomplish the invention so the "Progress of Science" can be advanced by others designing around the invention and/or using the technique once the patent expires.

Thus, the best mode requirement can be used to invalidate a patent where the inventor or assignee knew of a better mode of practicing the

117. CHISUM, *supra* note 25, at § 8.03.

118. *In re Warmerdam*, 33 at 3d 1354 (Fed. Cir. 1994).

119. *Id.* at 1360.

120. 35 U.S.C. § 112.

121. *Amgen, Inc. v. Chugai*, 927 F.2d 1200 (Fed. Cir. 1991).

invention at the time the application was filed and concealed it. The standard is a subjective standard, and if the inventor believed the disclosed method to be the best, knowledge of an objectively superior method is not grounds for invalidation. Decisions vary as to whether there must be evidence of intent to conceal.¹²²

As far as software is concerned, the best mode requirement applies as it does in other technological areas. The best mode requirement, however, may offer particularly fertile ground for a litigant seeking to invalidate a patent. Since many software patents are filed while the software itself is under development, programs may be undergoing change right up to and perhaps beyond the date of filing of the patent. In comparison with many structural inventions that are difficult to change during the course of development, software may be changed relatively quickly. The program may have changed in some regard between the time the patent attorney took the disclosure and the time the completed application was actually reviewed, signed, and filed. The inventors may not appreciate the importance of including the latest improvements.

Thus, as a defendant, it is well worth investigating whether a program was still under development at the date of filing to determine whether the inventors truly did include the best mode which was known when the patent was filed. Courts should not be expected to be sympathetic to patentees who have not lived up to their part of the *quid pro quo* of the patent system.

IV. CONCLUSION

This Comment provides a comprehensive review of the technological and legal bases for claiming software embodied in a computer readable medium. Despite being endorsed by the PTO's own guidelines, such claims still stand on somewhat tenuous legal ground. Nonetheless, solid technical and legal rationale is available to support such claims when the appropriate judicial rigor is applied. In the mean time, a practitioner who carefully crafts claims to closely track existing precedent and to avoid unduly limiting claim construction, can safely enjoy the utmost patent protection allowed by the prior art. Indeed, the Progress of Science continues as the patent systems embraces the endless stream of ones and zeroes that brings to us the Information Age.

Finally, the question posed must be answered. Are Beaugard's claims valid? Beaugard's claims likely are statutory under § 101, and

122. See, e.g., *Engel Industries, Inc. v. Lockformer Co.*, 946 F.2d 1528, 1533 (Fed. Cir. 1991) (listing "intentional concealment of the best mode" as "culpable conduct" which is ground for a finding of inequitable conduct). See also *Northern Telecom, Inc. v. Datapoint Corp.*, 908 F.2d 931, 941 (Fed. Cir. 1990) (holding that a showing of intentional concealment is not required for a best mode requirement violation).

should be adjudged as to their novelty based on the claim limitations reciting computer program steps. Nonetheless, an accused infringer would certainly not be raising a frivolous argument in alleging them to be invalid despite the program content. The *In re Lowry* rationale alone does not mandate that result for Beauregard's claim be upheld; however, a solid technical and legal argument, namely that software embodied in a computer readable medium is a functional component of a computer system, can be advanced in support of such claims.

*Jeffrey S. Draeger*¹²³

123. Mr. Draeger was awarded First Prize in the Intellectual Property Student Writing Competition sponsored by the Oracle Corporation.

