STATE STREET: LEADING THE WAY TO CONSISTENCY FOR PATENTABILITY OF COMPUTER SOFTWARE

CHRISTOPHER S. CANTZLER

INTRODUCTION

"Your competitors are using software patents to gain a business advantage, shouldn't you?"[1] This question greets visitors to a web site dedicated to software, internet, and computer patenting.[2] The site continues by claiming that "the most talked about, quoted, and debated patent case of 1998"[3] was *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*[4] Commentators have proclaimed that *State Street* is a "groundbreaking" decision[5] that sets "[s]eismic [p]recedent."[6] It has been hailed by "intellectual property attorneys everywhere . . . as a matter of tremendous importance"[7] because it "signals that software patents must be reckoned with."[8] The patent at issue in *State Street* covers what is essentially only a calculator used with mutual funds. The question, therefore, is why this case is so important.

The answer is that *State Street* has resolved the long-standing confusion as to *whether* computer software is generally patentable subject matter.[9] Since the United States Supreme Court denied certiorari on January 11, 1999, *State Street* has been the law of the land: computer software *is* pat-

---

1. Brian L. Longest, *Software Patent News* (visited Nov. 2, 1999) <http://www.softwarepatentnews.com/spn_february_1999.stm>.
2. *See id.*
3. *Id.*
4. 149 F.3d 1368 (Fed. Cir. 1998), *cert. denied*, 119 S. Ct. 851 (1999).
5. *See* Tony V. Pezzano, State Street *Court Case Opens World for Tech Patents*, SEC. INDUSTRY NEWS, Jan. 18, 1999, at 2.
6. *See* William T. Ellis & Aaron C. Chatterjee, State Street *Sets Seismic Precedent*, NAT'L L.J., Sept. 21, 1998, at B13.
7. Andrew B. Katz, 'State Street' *May Place Start-ups in Peril*, N.Y. L.J., Jan. 19, 1999, at C1.
8. *Id.*
9. *See id.*

entable subject matter. However, the battle over software patents is far from over.[10] The crucial issue in the wake of the *State Street* decision is determining the proper *scope* of computer software patents.

When new technologies develop, intellectual property law must adapt either to cover or not to cover the new inventions.[11] Intellectual property law encompasses patent, trademark, copyright, and trade secret law, including federal, state, statutory, and common law sources. Some inventions may be covered by more than one type of intellectual property law protection. Other inventors may be forced to choose between the types of law with which they will protect their creations. For reasons discussed later in this comment, patent law provides the strongest protection for most types of intellectual property.

However, despite numerous technological advances in the past century,[12] the statutory language of the Patent Act[13] (the "Act") pertaining to patentable subject matter has not changed significantly since the inception of the Act over 200 years ago.[14] Without current legislative guidance, courts and administrative agencies, including the United States Patent and Trademark Office (the "PTO"), have the difficult job of applying antiquated laws to cutting-edge technology, resulting in a confusing legal landscape.[15]

Advances in the field of computer technology, including both hardware[16] and software innovations,[17] have generated

---

10. *See id.*

11. *See, e.g.*, O'Reilly v. Morse, 56 U.S. (15 How.) 62 (1853) (upholding Morse's telegraph claim but declaring a claim covering the use of electromotive force to send messages of intelligible characters at any distance, which would have preempted later inventions such as the television and fax machines, invalid).

12. Computers, biotechnology, and the Internet quickly come to mind as examples of fields in which patenting might prove troublesome.

13. 35 U.S.C. §§ 1–376 (1994).

14. *See In re* Alappat, 33 F.3d 1526, 1552 (Fed. Cir. 1994) (Archer, C.J., concurring in part and dissenting in part) ("The terms used in §101 have been used for over two hundred years—since the beginning of American patent law—to define the extent of the subject matter of patentable invention.").

15. *See* Peter Weissman, *Computer Software as Patentable Subject Matter: Contrasting United States, Japanese, and European Laws*, 23 AIPLA Q.J. 525, 527 (1995).

16. *See* Gerald Sobel & Joseph Levi, *Protection of Computer Software*, *in* COMPUTER SOFTWARE PROTECTION, at 569, 575 (PLI Pat., Copyrights, Trademarks, & Literary Prop. Course Handbook Series No. 479, 1997) ("The term 'hardware' refers to the mechanical, magnetic and electrical devices which make up the computer.").

much of this legal confusion. Currently, the United States and most other countries provide copyright protection for computer programs.[18] However, as the size and economic value of the computer software industry have grown,[19] the demand for patent protection has increased.[20] Unfortunately, computer software does not fit neatly into any of the established categories of patentable subject matter.[21] Because of this poor fit, the courts have been reluctant to grant computer software patents for over thirty years,[22] as shown by numerous subject matter rejections of patent applications.[23]

This trend of subject matter rejections has reversed in recent years in both the PTO and the courts.[24] *State Street* swung the pendulum to the other extreme, holding that any-

---

17. *See id.* ("The term 'software' refers to all materials encompassing or describing the set of ordered instructions, or programs, which are used to direct the operations of computer 'hardware.'"). The terms "software" and "program" are often used interchangeably.

18. *See* Patricia A. Martone, *Intellectual Property Protection for Computer Software—As Copyright Protection Narrows, Can Patents Fill the Gap?*, *in* COMPUTER SOFTWARE PROTECTION, *supra* note 16, at 599, 612.

19. *See* Longest, *supra* note 1 ("IBM generates over $1 billion annually from the licensing of their intellectual property."); *see also* Keith E. Witek, *Developing a Comprehensive Software Claim Drafting Strategy for U.S. Software Patents*, 11 BERKELEY TECH. L.J. 363, 364 (1996).

20. Patent protection is, in many ways, stronger than copyright protection. *See* Carey R. Ramos & David S. Berlin, *Three Ways to Protect Computer Software*, 16 COMPUTER LAW. 16, 21 (1999) ("A valid patent provides the strongest possible protection for computer software."); *see also* Weissman, *supra* note 15, at 527.

21. *See* Weissman, *supra* note 15, at 529.

22. *See* ROBERT PATRICK MERGES, PATENT LAW AND POLICY 46 (2d ed. 1997) (discussing the first crop of computer software patent applications in the 1950s and early 1960s). "[T]he Patent Office met these with a uniform response: whatever software is, it is definitely *not* patentable subject matter." *Id.*

23. *See* Parker v. Flook, 437 U.S. 584 (1978) (holding that a computer program which updates alarm limits through a mathematical calculation was not patentable subject matter under section 101); Gottschalk v. Benson, 409 U.S. 63 (1972) (holding that a computer program which converted binary coded numbers into pure binary numbers was not patentable subject matter under section 101 of the Patent Act); *In re* Warmerdam, 33 F.3d 1354 (Fed. Cir. 1994) (holding that a computer program which created a specific data structure was not patentable subject matter under § 101); *see also* Witek, *supra* note 19, at 366.

24. *See, e.g.*, Diamond v. Diehr, 450 U.S. 175 (1981) (holding that a computer program which inventively applies a mathematical algorithm to a physical process or structure is patentable subject matter under section 101); *In re* Alappat, 33 F.3d 1526 (Fed. Cir. 1994) (holding that a computer program which transformed data through mathematical calculations to produce a smooth waveform display is patentable subject matter under § 101); *see also* David Bender, *Recent Developments in Software Patents*, *in* COMPUTER SOFTWARE PROTECTION, *supra* note 16, at 139, 152 .

thing "useful" is patentable subject matter.[25] In another opinion issued on the same day as *State Street*, Judge Clevenger of the Federal Circuit alluded to the *State Street* decision by stating, "patents are the backbone of much of the national economy, and, as this court has recently held, virtually anything is patentable."[26]

This dramatic holding has "opened the floodgates to the progressive use of . . . patents as a strong competitive weapon."[27] While this may be good news to patent attorneys, some restrictions are necessary. The *State Street* standard is so lenient that the PTO will be flooded with new patent applications that it is ill prepared to handle.[28] In addition, because the traditional subject matter rejection may be unavailable to patent examiners, the PTO may be tempted to issue patents too easily, without subjecting the applications to the full patentability analysis. This comment argues that, although the patent system provides the best choice of the available options for protecting computer software inventions, and although *State Street* was a step in the right direction, the current patent system needs further changes in order to meet the complex and unique needs of the computer software industry.

This comment begins with an introduction to computers and computer software in Part I, including an explanation of why computer software needs intellectual property protection. Part II then analyzes the various intellectual property alternatives to patent law protection of computer software. Part III asserts that patent law provides the most effective protection for computer software. Historically, court decisions involving

---

25. *See* State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1373 (Fed. Cir. 1998), *cert. denied*, 119 S. Ct. 851 (1999). (stating that computer software is useful if it "constitute[s] a practical application of an abstract idea"); *see also* Katz, *supra* note 7 ("The holding effectively eliminates the existing barriers to patentability for almost all types of computer software.").

26. Hughes Aircraft Co. v. United States, 148 F.3d 1384, 1385 (1998) (Clevenger, J., dissenting).

27. Ellis & Chatterjee, *supra* note 6.

28. *See* John T. Aquino, *Patently Permissive: PTO Filings Up After Ruling Expands Protection for Business and Net Software*, 85 A.B.A. J. 30 (May 1999).

> Already, the June 1998 decision appears to have set off a stampede to the patent office, which reports a 45 percent increase in the number of data processing and computer-related patents issued during its 1998 fiscal year . . . . Patent lawyers also cite anecdotal evidence that the numbers are way up for 1999.

Id.

patentable subject matter have been confusing at best. This case history, examined in Part IV, leads to the *State Street* decision, which is discussed in Part V. Part VI considers consequences of *State Street*, including the Federal Circuit's recent revision of *State Street* in *AT&T Corp. v. Excel Communications, Inc.*[29] Finally, Part VII outlines some possible solutions to the problems of patenting computer software inventions under the current system. The solutions include a proposed revision of section 101 of the Act in order to codify the *State Street* holding, a refocusing of the judicial inquiry away from subject matter suitability and towards overall patentability issues, and allocating additional resources to help the PTO handle computer software patent applications throughout the patenting process.

I. COMPUTER SOFTWARE AND ITS RELATIONSHIP TO INTELLECTUAL PROPERTY PROTECTION

Various forms of intellectual property protection have been applied to computer software inventions since their inception.[30] In order to understand the arguments for and against the various choices of intellectual property protection applicable to computer software, an explanation of the basic technology and terminology is necessary. Judge Sloviter of the Third Circuit provided a useful introduction to the world of computers in the landmark copyright case *Apple Computer, Inc. v. Franklin Computer Corp.*[31]

All computers have a central processing unit (the "CPU"), which is the "integrated circuit that executes programs."[32] Judge Sloviter explained that, "[i]n lay terms, the CPU does the work it is instructed to do."[33] The instructions for the CPU are contained in computer programs, which may be written in one of three levels of computer language.[34] High level language[35] "uses English words and symbols, and is relatively easy

29. 172 F.3d 1352 (Fed. Cir. 1999), *cert. denied,* 120 S. Ct. 368 (1999).

30. *See* Ramos & Berlin, *supra* note 20, at 16.

31. 714 F.2d 1240, 1243 (3d Cir. 1983). This case was one of the first cases to tackle the copyrightability of computer programs. *See* Katz, *supra* note 7.

32. *Apple*, 714 F.2d at 1243.

33. *Id.*

34. *See id.*

35. High level languages include, for example, BASIC or FORTRAN.

to learn and understand."[36] Assembly language is a "somewhat lower level language [than high], which consists of alphanumeric labels (e.g., 'ADC' means 'add with carry')."[37] Statements written in high or assembly language are referred to as "source code."[38] Machine language, the lowest level of computer language, is a "binary language using two symbols, 0 and 1, to indicate an open or closed switch (e.g., '01101001' means, to [an] Apple [computer], add two numbers and save the result)."[39] Statements written in machine language are referred to as "object code."[40] Importantly, the "CPU can only follow instructions written in object code."[41] However, since object code is very difficult for humans to understand, computer programs are usually written in source code.[42] Programs written in source code "can be converted or translated by a 'compiler' program into object code for use by the computer."[43]

Computer programs can be stored on a variety of storage devices, including floppy disks, CD-ROMs, and hard drives.[44] Computer programs must be transferred from the storage device into the computer's "working memory," called Random Access Memory or RAM, in order for the instructions to be given to the CPU. The instructions can then be used to perform a specific task, like word processing or playing a game,[45] or they may be used to manage the internal functions of the computer.[46] Computer programs are "generally distributed only in their object code version stored on a memory device,"[47] although

---

36. *Apple*, 714 F.2d at 1243.
37. *Id.*
38. *See id.*
39. *Id.*
40. *See id.*
41. *Id.*
42. *See* 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, 142 (1998) ("Generally stated, the less comprehensible software is to a person[,] the more directly useable it is by a computer."); *see also Apple*, 714 F.2d at 1243.
43. *Apple*, 714 F.2d at 1243.
44. *See id.*
45. Computer programs of this kind are called "application programs." *See id.*
46. Computer programs of this kind are called "operating system programs." *See id.* The distinction between application and operating system programs is unimportant to this comment.
47. *Id.*

with the emergence of the internet and computer networks, this general pattern is changing.[48]

Software needs some form of legal protection because the "behavior and other industrial design elements of programs are often expensive to develop and inexpensive to copy."[49] This allows competitors to develop "clones"—imitation programs that are functionally indistinguishable from the originals—quickly and cheaply, and then sell them at a lower price.[50] These clones reduce profits for the original inventor and thereby reduce the incentive to develop new software inventions, because the original inventor does not recoup the high initial research and development costs.[51] In addition, the original programs themselves are easily copied and distributed to multiple users, further reducing profits for the original developer. Both tactics are forms of software piracy. Despite the fact that software companies are working on ways to reduce this copying through the use of encoding and other security measures, the impact of piracy on the software industry is enormous. The International Intellectual Property Alliance estimated 1995 losses to United States software companies as a result of piracy at more than ten billion dollars.[52]

In addition to protecting against unauthorized copying, it is also crucial to have a predictable system of intellectual property protection for computer software in order to clearly define the legal rights of competitors in the software market.[53] By establishing the extent to which companies may use elements of existing software in their new products, the incentives for innovation, a cornerstone of patent law, will be maximized.[54] This rationale is especially important given the increased use of "object-oriented" software development, through which sec-

---

48. Networks in this context are several computers linked together so that programs and data can be shared between them, as in an office environment. *See* Chiappetta, *supra* note 42, at 153.

49. Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2312 (1994).

50. *See id.*

51. *See id.* This problem is known in law and economics as the "free rider problem." *See* Ramos & Berlin, *supra* note 20, at 16.

52. *See* Alice Rawsthorn, *Piracy 'Costs US Industries $14.6bn,'* FIN. TIMES (London), Feb. 10, 1997, at 6.

53. *See* Ramos & Berlin, *supra* note 20, at 17.

54. *See infra* notes 111–15 and accompanying text for a discussion of the incentive rationale of the patent grant; *see also* Ramos & Berlin, *supra* note 20, at 17.

tions of code are designed and used interchangeably to perform simple, specific "building block" tasks that are routinely used in several different types of programs.[55] As a result, some form of intellectual property protection is needed for computer software.

II. ALTERNATE METHODS OF INTELLECTUAL PROPERTY PROTECTION FOR COMPUTER SOFTWARE INVENTIONS

Computer software presents some interesting, if not unique, problems for intellectual property protection.[56] These difficulties have led commentators to recommend several different kinds of intellectual property protection for computer software, including copyright, trade secret, and patent protection.[57] The following brief introduction to the alternate methods of intellectual property protection of computer software explains why computer software can be protected best by our current, albeit slightly modified, patent system.

A. Copyright Protection of Computer Software

Copyright protection of computer software is often mentioned as the complement and most likely alternative to patent protection of computer software.[58] Copyrights give protection to the original expression of an idea,[59] rather than to the idea itself.[60]

---

55. *See* Ramos & Berlin, *supra* note 20, at 16; Samuelson et al., *supra* note 49, at 2322, 2329.

56. *See* Ramos & Berlin, *supra* note 20, at 16 ("Since these traditional forms of protection were developed to protect industrial inventions and artistic works, they have sometimes proven an awkward fit for computer programs.").

57. *See id.*

58. *See* Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) (establishing the abstraction-filtration-comparison test for determining copyright infringement of computer software); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983) (holding computer program in object code is copyrightable subject matter). *See generally* 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).

59. *See* 17 U.S.C. § 102(a) (1994) ("Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression . . . .").

60. *See* 17 U.S.C. § 102(b) (1994) ("In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery . . . ."); *see also* Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1234 (3d Cir. 1986) ("Unlike a

The actual code, either source or object, comprising a piece of computer software is protected as a "literary work" by the Copyright Act of 1976[61] since it is the expression of how to instruct a computer to perform a certain task.[62] This is similar to the protection afforded the text of a novel, and exists whether or not the code is understandable to the average human reader.[63] The courts consistently have upheld this protection.[64]

Given appropriate subject matter, the copyright registration process is not burdensome and is even to some degree optional. It consists of a twenty dollar fee[65] and a simple registration form, both of which are required only if the copyright is to be sued upon in an infringement action.[66] Irrespective of a registration, copyright protection is automatic and lasts for seventy years from the death of the author or ninety-five years after publication of an anonymous work or a work made for hire.[67] Copyright law gives longer protection than patent law,[68] but since the profitable life of computer programs tends to be on the order of months or years instead of decades, this longer protection is not significant.[69]

The primary drawback with existing copyright protection for computer programs is the limited scope of the protection. As noted above, a fundamental principle of copyright law is that an author's expression is protectable, but the underlying idea of a work is not protectable.[70] There are usually a number of different ways to tell a computer to do a specific task, and computer programmers can often change the exact wording of

---

patent, a copyright gives no exclusive right to the art disclosed; protection is given only to the expression of the idea—not the idea itself." (quoting Mazer v. Stein, 347 U.S. 201, 217 (1954))).

61. *See* 17 U.S.C. §§ 101–810, 1001–1010 (1994).

62. *See* 17 U.S.C. § 101 (1994) ("A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.").

63. *See* Ramos & Berlin, *supra* note 20, at 18.

64. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983); *see also* Martone, *supra* note 18, at 601.

65. *See* 17 U.S.C. § 708(a)(1) (1994).

66. *See* 17 U.S.C. § 411(a) (1994).

67. *See* 17 U.S.C. § 302(a), (c) (1994).

68. Patent law extends protection for twenty years from the date of filing of the patent application. *See* 35 U.S.C. § 154(a)(2) (1994).

69. *See* Robert C. Scheinfeld & Gary M. Butter, *Using Trade Secret Law to Protect Computer Software*, 17 RUTGERS COMPUTER & TECH. L.J. 381, 405–06 (1991).

70. *See* Baker v. Selden, 101 U.S. 99, 104 (1879).

the code and still achieve the same functionality.[71] Since the software producer is usually far more concerned with the function of the program than its exact wording, these minor changes effectively copy the original program without infringing the copyright.[72] This greatly limits the effectiveness of copyright protection.[73]

Consequently, courts initially attempted to expand copyright protection to include the non-literal aspects of the code—like the organization, structure, and sequence of the program—which are more likely to produce the functionality of the program.[74] Recently, however, courts have restricted copyright protection for computer programs due to a fear that overprotecting the utilitarian aspects of the software will grant a patent-like monopoly without subjecting the invention to the rigors of the patent application process.[75] In addition, commentators have noted that the broader protection "grants an unreasonable advantage to the authors of existing software and unduly restricts future innovation in the field."[76] While there has been disagreement as to the exact boundary between literal and non-literal aspects of computer code,[77] it appears

---

71. *See* Martone, *supra* note 18, at 602; *see also* Ramos & Berlin, *supra* note 20, at 17.

72. *See* Martone, *supra* note 18, at 602; *see also* Ramos & Berlin, *supra* note 20, at 17.

73. *See* Martone, *supra* note 18, at 602.

74. *See id.*

75. *See id.* at 614.

> It is particularly important to exclude methods of operation and processes from the scope of copyright in computer programs because much of the content of computer programs is patentable. Were we to permit an author to claim copyright protection for those elements of the work that should be the province of patent law, we would be undermining the competitive principles that are fundamental to the patent system.

Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1541 n.21 (11th Cir. 1996); *see also* Part III, *infra*, for a discussion of the various hurdles a patent application must pass before a patent is granted.

76. Ramos & Berlin, *supra* note 20, at 18.

77. *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided court*, 516 U.S. 233 (1996) (holding that a spreadsheet menu command hierarchy was a "method of operation" and therefore non-copyrightable subject matter under § 102(b) of the Copyright Act). *But see* Mitel, Inc. v. IQTel, Inc., 124 F.3d 1366, 1372 (10th Cir. 1997).

> Section 102(b) does not extinguish the protection accorded a particular expression of an idea merely because that expression is embodied in a method of operation at a higher level of abstraction. Rather, sections 102(a) & (b) interact to secure ideas for [the] public domain and to set

that "the useful behavior generated when programs are in operation and the industrial design responsible for producing this behavior" are ill-suited to copyright protection.[78]

A further complication unique to computer software is that the same computer language (code) can serve multiple functions for the same program.[79] The code may be used to express the design of the program to a human reader, much like the text of a book. This use is the traditional province of copyright law. Yet the same code may be the actual *tool* which is used to produce a certain result in the computer through the instruction of the CPU. This use is akin to a traditional invention and is therefore likely the province of patent law.[80] Since the specific text of a computer program is often far more valuable for the actual results it produces in a computer than for its literary quality, this greatly limits the value of copyright protection.[81]

A significant downside to copyright protection is that, unlike patent law, it does not protect against independent creation.[82] Since companies often have similar goals, like developing a better word processing program, independent creation is often a distinct possibility. Moreover, the plaintiff bears the heavy burden of proving copying instead of independent creation at trial, thereby reducing the chances of recovering against a potential infringer.[83]

---

apart an author's particular expression for further scrutiny to ensure that copyright protection will "promote the . . . useful Arts."

Id. (quoting U.S. CONST. art. 1, § 8, cl. 8).

78. Samuelson et al., *supra* note 49, at 2310; *see also* Ramos & Berlin, *supra* note 20, at 18–19.

79. *See* Samuelson et al., *supra* note 49, at 2316.

80. *See* Chiappetta, *supra* note 42, at 141–42 (discussing at length this interesting distinction regarding the multiple "uses" of computer language and distinguishing this from the "idea/expression" dichotomy); *see also* Karjala, *supra* note 58, at 42.

81. *See* Chiappetta, *supra* note 42, at 142.

82. Without protection from independent creation, if one person creates an identical program as someone else without copying from the first author, the second author is not liable to the first—he is not an "infringer." *See infra* text accompanying note 141 for a discussion of independent creation in the patent law context.

83. *See* Bender, *supra* note 24, at 147 ("In [copyright] litigation it is often difficult to show copying. In patent litigation, it is unnecessary to do so."); *cf.* Nora M. Tocups & Robert J. O'Connell, *Patent Protection for Computer Software*, 14 COMPUTER LAW., Nov. 1997, at 18, 19 ("[T]he broad protection of the patent protects against those who independently develop a software product that includes the invention in the patent, regardless of whether the intent was to infringe or not.").

Another disadvantage of copyright protection is that the common practice[84] of reverse engineering of computer programs has thus far been protected by the fair use doctrine.[85] The fair use doctrine exempts a user from infringement liability in cases of limited forms of copying that are determined to be in the public interest.[86] Reverse engineering has been held to be a fair use because it helps produce competitive products.[87] Reverse engineering is the process by which a user disassembles copyrighted object code for the purpose of figuring out how the program works.[88] The process is meant to uncover non-copyrightable elements of the program, although it usually requires some temporary copying of the original software.[89] As long as it continues to be lawful, reverse engineering may only be prevented through restrictions imposed under contractual license agreements.[90] These disadvantages make copyright law an inferior choice for protecting computer software.

B. Trade Secret Protection of Computer Software

Trade secret protection is also a possible alternative to patent protection for computer software.[91] Trade secret law protects valuable information that is not generally known to competitors and gives the owner a competitive advantage in the marketplace.[92] Computer software certainly can qualify for this protection.[93] One advantage of this protection is that the

---

84. *See* Ramos & Berlin, *supra* note 20, at 19. Ramos and Berlin note that reverse engineering tends to be done not by individuals who would pirate a product wholesale, but by rivals with the resources and expertise to successfully complete the somewhat difficult task. *See id.* at 23.

85. *See* Sega Enter., Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992); Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992).

86. *See* 17 U.S.C. § 107 (1994).

87. *See Sega*, 977 F.2d at 1514 (limiting its holding to circumstances where there is a "legitimate reason for doing so and when no other means of access to the unprotected elements exists").

88. *See* Ramos & Berlin, *supra* note 20, at 19.

89. *See id.*

90. *See id.* at 20.

91. *See generally* Scheinfeld & Butter, *supra* note 69.

92. *See* RESTATEMENT (THIRD) OF UNFAIR COMPETITION § 39 (1995). Trade secret law is governed by the law of the individual states. *See* Ramos & Berlin, *supra* note 20, at 20.

93. *See* Data Gen. Corp. v. Grumman Sys. Support Co., 36 F.3d 1147 (1st Cir. 1994) (holding that confidential contents of a computer program distributed

trade secret itself need not meet the novelty and nonobviousness thresholds required by the Patent Act.[94] Furthermore, trade secrets can last forever, as long as they do not become well known.[95] Computer software, however, can become well known through reverse engineering or widespread distribution.

Unfortunately, the value of a computer program often increases with widespread distribution, and this conflicts with the requirements of trade secret law.[96] Confidentiality agreements and other safeguards may be required with both employees and customers in order to keep the trade secret from becoming well known. Furthermore, although patent prosecution is kept secret,[97] the disclosure requirement for patent applications voids any trade secrets contained in the application once the patent issues and is published.[98] Therefore, "both a patent and a trade secret in the same subject matter cannot coexist."[99]

Another significant drawback of trade secret protection is that it does not protect against independent creation.[100] Fortunately for the inventor, trade secret law requires the accused misappropriator to prove independent creation at trial.[101] The independent creator, however, may seek and obtain a patent, which would exclude the first inventor who had relied on trade secret protection.[102]

C. Other Forms of Protection for Computer Software

Trademark protection is available for computer programs, but it contains no protection for the functional aspects of computer software, and therefore has very limited utility in the

---

only in object code format qualified as trade secrets); *see also* Scheinfeld & Butter, *supra* note 69, at 382.

94. *See infra* Part III for a discussion of these requirements.

95. *See* Scheinfeld & Butter, *supra* note 69, at 405.

96. *See id.* at 407.

97. *See* 35 U.S.C. § 122 (1994) (providing for the confidentiality of patent applications).

98. Upon issuing, patent applications are published. *See* 35 U.S.C. § 11(a) (1994) ("The Commissioner may print, or cause to be printed . . . [p]atents, including specifications and drawings, together with copies of the same."); *see also infra* notes 117–21 and accompanying text for a discussion of the disclosure requirement for patents.

99. Scheinfeld & Butter, *supra* note 69, at 399.

100. *See id.* at 406.

101. *See id.* at 395 n.58.

102. *See id.*

computer software industry.[103] However, there are other mechanisms that may protect computer software inventions.

The software market itself rewards the originators of computer software inventions for getting a head start on the competition, and thereby encourages future inventions. By being the first to develop a successful new product, a company can expect to earn customers and make money before the competition can develop a competing product.[104] Furthermore, name recognition for being the pioneer in the field will continue after competing products enter the market.[105]

A stronger mode of protection is the shrinkwrap license used by mass-marketed software and the click-on license used for software distributed over the internet.[106] Legally, both function the same way, in that breaking open the software shrinkwrap, or clicking "yes" before downloading a program from the internet, constitutes acceptance of the terms of the accompanying license under state contract law.[107] These licenses "typically [prohibit] the user from copying and reverse engineering the licensed program."[108] Initially, there was some debate over the validity of shrinkwrap licenses, but the growing consensus is that they are enforceable, provided that the consumer has the opportunity to review the license and decline or return the software if the terms are not acceptable.[109] Despite being enforceable, however, these licenses can be difficult to police, and infringement of the provision tends to be difficult to prove.[110]

---

103. Trademark law protects against anyone who uses a designation that is likely to cause some confusion in the minds of the consumer regarding the source of the product. *See* 15 U.S.C. § 1114(1) (1994). This means that the Microsoft® name, the Windows® logo, and the "intel inside®" designation would all be protected from competitors using similar designations as long as the trademark owner could show that consumers were confused by the competitor's designation.

104. *See* MERGES, *supra* note 22, at 156 (discussing the head start advantage).

105. *See id.*

106. *See* Ramos & Berlin, *supra* note 20, at 20.

107. *See id.* at 21.

108. *Id.*

109. *See* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996) (indicating that a shrinkwrap license was enforceable). A proposed new Article 2B of the UCC would codify the enforceability of shrinkwrap licenses if adopted. *See* UCC Article 2B—Licenses § 2B-111 (Tentative Draft 1998).

110. *See* Ramos & Berlin, *supra* note 20, at 22.

Given the need for intellectual property protection of computer programs and the significant limitations of the various forms of protection already discussed, the next section analyzes patent law protection.

III. PATENT LAW AND ITS APPLICATION TO COMPUTER SOFTWARE

This Part begins with a brief discussion of patent law generally, including the policies behind the Patent Act and the procedures involved in obtaining a patent. A discussion of the advantages and disadvantages of patenting computer software and a comparison to other forms of intellectual property protection concludes that patent law is the best form of protection for computer software.

A. Patent Law Generally

The United States patent system is rooted in the Constitution, which empowers Congress to "promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."[111] Given the pro-industrial sentiment present during the founding of the nation, Congress acted quickly upon the constitutional grant of power to adopt the first United States Patent Act in 1790.[112] Under the Act, the government essentially grants a twenty year[113] monopoly to an inventor in exchange for a complete disclosure of the invention to the public.[114] The purpose of this exchange is to provide incentive for growth in the useful arts, resulting in "overall economic growth and a general increase in prosperity."[115]

In order to obtain a patent, an inventor must file an application with the PTO. The PTO assigns the application to a

---

111. U.S. CONST. art. 1, § 8, cl. 8.

112. *See* MERGES, *supra* note 22, at 8–9.

113. *See* 35 U.S.C. § 154(a)(2) (1994).

114. The Supreme Court has expressly rejected the notion that an inventor has a natural or moral right to possess what he or she invents. *See* Diamond v. Chakrabarty, 447 U.S. 303, 308 (1980).

115. Chiappetta, *supra* note 42, at 98, 130 (explaining that the constitutional term "useful arts" is equivalent to the present day term "technological arts").

patent examiner, who handles the application through the entire process.[116] The application must contain a section called the "specification," which contains a detailed description of the invention and the written "claims."[117] This specification must describe the invention "in such full, clear, concise, and exact terms as to enable any person skilled in the art" to make or use the invention.[118] These two related requirements of section 112 are called the "description" and "enablement" requirements.[119] In theory, the section 112 requirements are the first hurdle in the application process,[120] since it is at this step that the actual claimed "invention" is defined.[121]

In addition to the section 112 requirements, the PTO will also reject an application if the claimed invention fails to satisfy the subject matter requirement,[122] the novelty requirement,[123] or the nonobviousness requirement.[124]

This comment is primarily concerned with the statutory grant of patentable subject matter found in section 101 of the Act: "Whoever 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."[125] To be patentable, an invention must fall within one of the four categories of process, machine, manufacture, or composition of matter, although it is not necessary for the inventor to designate into which category the invention falls.[126] The four categories can be divided into two broad headings: method or "proc-

116. *See* 35 U.S.C. § 111(a)(1) (1994).

117. *See* 35 U.S.C. § 112 para. 2 (1994) ("The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.").

118. 35 U.S.C. § 112 para. 1 (1994).

119. *See* Chiappetta, *supra* note 42, at 104.

120. *Cf. In re* Bergy, 596 F.2d 952, 960 (C.C.P.A. 1979) ("The first door which must be opened on the difficult path to patentability is § 101.").

121. *See* Chiappetta, *supra* note 42, at 104.

122. *See* 35 U.S.C. § 101 (1994).

123. *See* 35 U.S.C. §§ 102(a), (e) (1994).

124. *See* 35 U.S.C. § 103 (1994 & Supp. III 1997).

125. 35 U.S.C. § 101 (1994).

126. *See* State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1372 (Fed. Cir. 1998), *cert. denied*, 119 S. Ct. 851 (1999) ("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 . . . ."); *see also* Weissman, *supra* note 15, at 529.

ess," and object or "apparatus," encompassing the last three statutory categories.[127] Due to the unique nature of computer software, only the composition of matter category has not been used to describe computer programs.[128]

Despite the broad categories of patentable subject matter discussed above, the Supreme Court has determined three categories of subject matter which do not fall within one of the allowed categories: "laws of nature, natural phenomena, and abstract ideas."[129] In addition, lower courts have noted two other judicially-created exceptions, the "mathematical algorithm" exception[130] and the "business method" exception.[131] As discussed below, the *State Street* case properly casts serious doubt on the status of these two lower court exceptions.

The purpose of the four statutory categories is to require inventions to have some tangible "real world" value, rather than being just ideas or concepts.[132] There is a concern that claims describing unpatentable subject matter will be "smuggled past the § 101 border guards by expressing them in software 'technology' terms."[133] This concern has been very strong historically, and the "vast majority of . . . software cases involved only § 101."[134]

Once a patent application passes the subject matter test, it must still pass two further tests not required in a copyright or

---

127. *See* Chiappetta, *supra* note 42, at 100 n.45.

128. *See id.*

129. Diamond v. Diehr, 450 U.S. 175, 185 (1981).

130. The Supreme Court defined an "algorithm" as a "procedure for solving a given type of mathematical problem." Gottschalk v. Benson, 409 U.S. 63, 65 (1972). However, the Court has also admitted that the "term 'algorithm' is subject to a variety of definitions." *Diehr*, 450 U.S. at 186 n.9. The Federal Circuit recently quipped that "[c]ourts have used the terms 'mathematical algorithm,' 'mathematical formula,' and 'mathematical equation,' to describe types of non-statutory mathematical subject matter without explaining whether the terms are interchangeable or different." AT&T Corp. v. Excel Communications, Inc., 172 F.3d 1352, 1356 (Fed. Cir.), *cert. denied*, 120 S. Ct. 368 (1999).

131. This means simply that methods of doing business were thought to be unpatentable subject matter. *See, e.g., State Street*, 149 F.3d at 1376 & n.13.

132. *See* Tocups & O'Connell, *supra* note 83, at 19. This requirement ensures that the invention falls within the "useful arts."

133. Chiappetta, *supra* note 42, at 100. Chiappetta also argues that both the mathematical algorithm test in *Gottschalk* and the FWA test, described *infra* at text accompanying note 200, were developed in an attempt to identify overly broad software claims which "involved no more than claims to 'laws of nature.'" *Id.* at 100 n.42; *see also In re* Schrader, 22 F.3d 290, 295 (Fed. Cir. 1994).

134. Chiappetta, *supra* note 42, at 100 n.46.

trade secret analysis, "novelty" and "nonobviousness."[135] Both tests ensure that real "progress" is made in the useful arts. The novelty requirement disqualifies any patent application that describes an invention that was already known to society.[136] The nonobviousness requirement extends the novelty requirement to disqualify not only those patent applications whose exact inventions were already known, but also those applications whose inventions would have been obvious "to a person having ordinary skill in the art," given the pre-existing state of knowledge in the field of the invention (known as the "prior art").[137] Only after an application has cleared all of these hurdles will a patent finally be granted.

B. Advantages and Disadvantages of Patent Protection of Computer Software

Patent law is "quickly becoming the most powerful tool in protecting appropriately selected software."[138] This section highlights the reasons why patent protection is preferable to other types of intellectual property protection for computer software.

1. Advantages

Unlike copyright and trade secret law, patent law allows the patent owner to exclude others from making, using, selling, or offering to sell the claimed invention for the twenty-year life

---

135. *See* Ramos & Berlin, *supra* note 20, at 17.

136. *See* 35 U.S.C. § 102(a) (creating an exception to patent eligibility when "the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for patent . . . .").

137. The statute reads:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences 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.

35 U.S.C. § 103(a) (1994); *see also* Graham v. John Deere Co., 383 U.S. 1, 13–19 (1966) (setting out the basic test for nonobviousness).

138. Scheinfeld & Butter, *supra* note 69, at 397–98.

of the patent.[139] There is no fair use doctrine to allow reverse engineering.[140] More importantly, this right to exclude applies even if the alleged infringer developed the invention independently without copying or misappropriating.[141] There is no intent or knowledge requirement to the infringement liability.[142] These advantages are extremely important in infringement litigation, where it can be particularly difficult to show copying—as required under copyright law—or misappropriation—as required under trade secret law—instead of independent creation. Patent protection is therefore more attractive than either of these alternatives because the greater chances for success at trial provide greater confidence in the scope and validity of the protection.[143]

Although the twenty-year term of a patent is shorter than the term of protection afforded by the copyright law, the rapid pace of technological innovations today tends to make the difference immaterial for most software inventions.[144] Additionally, the validity of a patent does not depend on secrecy like the validity of a trade secret does. This saves time and effort in dealing with customers and employees, since confidentiality agreements and other safeguards that protect trade secrets are not required. Therefore, proliferation of the software, which tends to increase the value of the program, is permitted.[145]

One policy-based advantage that is not often mentioned by commentators is that all patent appeals, unlike copyright or trade secret appeals, are handled by the Court of Appeals for the Federal Circuit. This provides for national uniformity, which in turn results in greater predictability in the law and how it will be applied. Greater predictability for the protection of innovation increases the incentive to innovate.

A practical advantage of patents is that even small companies can use patents to protect themselves from much larger

---

139. *See* 35 U.S.C. § 271(a) (1994) ("Except as otherwise provided in this title, whoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent.").

140. *See* Ramos & Berlin, *supra* note 20, at 21.

141. *See id.*

142. *See id.*

143. *See* Bender, *supra* note 24, at 147.

144. *See* Ramos & Berlin, *supra* note 20, at 23.

145. *See id.* at 17.

competitors through the threat of large monetary judgments.[146] A noteworthy example is the 1994 decision in the case of *Stac Electronics v. Microsoft*.[147] Stac owned a patent for a data compression program.[148] Microsoft considered licensing the program for use with its operating system software, MS-DOS 6.0, but decided instead to use its own data compression software.[149] Stac sued for patent infringement in the United States District Court for the Central District of California, where the jury ultimately awarded $120 million to Stac upon finding that Microsoft had infringed Stac's valid patent.[150] The case was subsequently settled.[151]

The advantages of patent protection outlined above tend to make patents very attractive for licensing as well.[152] Indeed, one commentator noted that "[p]atents provide stronger protection than copyrights, are a strong deterrent against reverse engineering, and are therefore viewed seriously by the software industry. The best evidence of this attitude is the level of patent cross-licensing that has occurred in recent years—and continues to occur—between major software producers."[153]

2. Disadvantages

There are, however, several disadvantages with patenting computer software under the current system. Although some of these disadvantages are inherent in the patent system, the unique nature of computer software has added to the difficulties. Part IV of this comment shows that the PTO and the courts have been historically reluctant to grant or enforce patents covering software inventions due to a perceived lack of patentable subject matter under section 101 of the Patent Act. Parts V and VI of this comment show that the *State Street* de-

---

146. *See id.* at 22.

147. No. CV-93-413-ER (C.D. Cal. May 13, 1994), *appeal voluntarily dismissed,* 38 F.3d 1222, *cited in* Ramos & Berlin, *supra* note 20, at 20 n.19.

148. *See* Ramos & Berlin, *supra* note 20, at 22.

149. *See id.*

150. *See id.* The case also demonstrates the important point that different forms of intellectual property protection for computer software can arise in the same dispute. Microsoft was awarded $13.6 million by the same jury on a counterclaim for a trade secret misappropriation by Stac. *See id.*

151. *See id.*

152. *See* Bender, *supra* note 24, at 147.

153. Ramos & Berlin, *supra* note 20, at 23.

cision may have finally solved at least this specific problem with software patenting.

Still, the barriers that any patent application must face in order to be issued remain significant. Unlike trade secret and copyright protection, patent protection is only available to those inventions that are so novel and so significantly different from all other similar inventions as to be nonobvious to those in the field. One trait of software inventions is that they tend to be incremental in nature, meaning that they build upon previous programs through minor improvements.[154] Since patent law requires substantial, nonobvious contributions to the public domain in exchange for the traditionally disfavored monopoly, many programs that would be commercially successful may not meet the requirements for a valid patent.[155]

Also, the time spent getting a computer software patent application to issue can exceed two years, which may be longer than the commercial life span of the software itself.[156] This time period is on average five months longer for software applications than for other kinds of inventions, mostly due to the complexity required to avoid section 101 rejections.[157] However, all patent applications tend to be lengthy processes due to the technical nature of the subject matter and the heavy workloads of the patent examiners.[158] The length and complexity of the process makes applying for a computer software patent expensive.[159] This investment is somewhat risky, since there is no guarantee that the patent will contain valuable claims, or even issue at all, after attempting to clear all of the statutory hurdles.[160]

Even after a patent issues, patent infringement litigation can be troublesome. After suing a potential infringer, the pat-

---

154. *See* Samuelson et al., *supra* note 49, at 2346.

155. *See* 35 U.S.C. § 103 (1994); *see also supra* note 137 and accompanying text for discussion of nonobviousness. This problem with incremental inventions also occurred with semiconductor chips. *See infra* notes 271–74 and accompanying text for a discussion of Congress's solution.

156. *See* Scheinfeld & Butter, *supra* note 69, at 399.

157. *See* Bender, *supra* note 24, at 148.

158. *See id.*

159. *See* Katz, *supra* note 7 (stating that the average fee for simply filing a computer software patent application is $7500, and that the cost of litigating a patent infringement action through trial and appeal is often over one million dollars).

160. *See* Scheinfeld & Butter, *supra* note 69, at 399.

ent owner usually will not be granted preliminary injunctive relief, unlike in trade secret and copyright litigation.[161] At trial, it is likely that the patent owner will have to defend the validity of the patent. Although ostensibly the burden of proving the patent's invalidity is on the defendant, the patent owner must bear the double burden of defending the validity of the patent while trying to show infringement by the defendant.[162] This is a difficult task, and the plaintiff's patent is often severely restricted or declared invalid.[163] Due to the fair chances of success for the alleged infringer, patent infringement suits are rarely resolved through settlement or dismissal.[164] Thus, the time and expense involved in such suits is quite high in comparison to other kinds of civil suits.[165]

Weighing the advantages and disadvantages listed above, a "valid patent provides the strongest possible protection for computer software."[166] Commentators have noticed a significant increase in the number of software patents issued in the last ten years, indicating that computer software developers also believe that patents offer the best protection available.[167] However, the courts and the PTO have not always shared this enthusiasm for software patents.

IV. THE CASE LAW DEALING WITH PATENTABLE SUBJECT MATTER UNDER SECTION 101

As shown above, patent law provides the strongest intellectual property protection for computer software. Consequently, there has been no shortage of cases dealing with computer software patent applications, most of which discuss patentable subject matter under section 101 of the Patent Act. This Part

---

161. *See* Bender, *supra* note 24, at 148.

162. *See* Scheinfeld & Butter, *supra* note 69, at 400–01.

163. *See* Ramos & Berlin, *supra* note 20, at 17 (stating that "twenty years ago, two out of three patents examined by the courts were held to be invalid—despite the fact that an issued patent enjoys a statutory presumption of validity."). The authors note that since the establishment of the Federal Circuit, seen to be a strong supporter of the rights of patent holders, the percentage of patents held valid after court review has doubled. *See id.*

164. *See* Bender, *supra* note 24, at 148.

165. *See id.*

166. Ramos & Berlin, *supra* note 20, at 21.

167. *See* Chiappetta, *supra* note 42, at 181 nn.19–20 (listing a number of web sites containing data on the trend).

reviews the major cases leading up to the *State Street* decision. The first section discusses the legal environment before the Supreme Court addressed the problem. The next section analyzes the three crucial Supreme Court cases. The last section deals with the Federal Circuit's attempts to incorporate the Supreme Court's guidance into the problem of computer software as patentable subject matter.

A. *Pre-*Benson*: Before the Supreme Court Entered the Fray*

In 1968, the Court of Custom and Patent Appeals (the "CCPA")[168] decided the first significant computer software case, *In re Prater*.[169] This case and other early cases looked promising to software patent applicants, as the CCPA was willing to reverse a reluctant PTO regarding patentable subject matter.[170]

B. *The Supreme Court Trilogy of Cases*

The Supreme Court has addressed the issue of computer software inventions as patentable subject matter only three

---

168. The Court of Appeals for the Federal Circuit ("CAFC") took over jurisdiction of all appeals from the Federal District Courts relating to patents as well as most appeals from PTO decisions from the CCPA on October 1, 1982. *See* Federal Courts Improvement Act of 1982, Pub. L. No. 97-164, 96 Stat. 25 (codified as amended at 28 U.S.C. § 41 (1994)). 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).

169. 415 F.2d 1393, 1406 (C.C.P.A. 1969) (reversing rejection, based on obviousness, of a claim for an apparatus which analyzes spectroscopic data because "one not having knowledge of appellants' discovery simply would not know what to program the computer to do.").

170. *See In re* Musgrave, 431 F.2d 882 (C.C.P.A. 1970).

> 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, 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.

Prater, 415 F.2d at 1403 n.29.

times.[171] The first decision, *Gottschalk v. Benson*,[172] severely curtailed optimism for software patents.

1. *Gottschalk v. Benson*

In *Benson*, the invention at issue was a method for converting binary coded decimal numerals into pure binary numbers.[173] The Court determined that the claimed method was an algorithm that was not directed to any particular application, so that granting a "patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself."[174] Noting that it "is conceded that one may not patent an idea,"[175] the Court determined that since the process did not transform or reduce one thing into another, the process was just an idea and therefore unpatentable subject matter under section 101.[176] Although *Benson* did not preclude the patentability of computer software,[177] it suggested that "all computer programs were nothing more than nonstatutory mathematical algorithms."[178]

2. *Parker v. Flook*

The next Supreme Court case, *Parker v. Flook*,[179] reinforced the *Benson* mathematical algorithm preemption approach.[180] *Flook* discussed patentable subject matter under

---

171. *See* Diamond v. Diehr, 450 U.S. 175 (1981); Parker v. Flook, 437 U.S. 584 (1978); Gottschalk v. Benson, 409 U.S. 63 (1972).

172. 409 U.S. 63 (1972).

173. *See Benson*, 409 U.S. at 64.

174. *Id.* at 72.

175. *Id.* at 71.

176. *See id.* at 70–72.

177. *See id.* at 71 ("It is said that the decision precludes a patent for any program servicing a computer. We do not so hold.").

178. Weissman, *supra* note 15, at 532 (citing *Benson*, 409 U.S. at 71, where the Court also stated that the algorithm had "no substantial practical application except in connection with a digital computer"). The PTO interpreted the *Benson* decision to mean that computer software inventions were basically unpatentable. *See 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).

179. 437 U.S. 584 (1978).

180. *See* Chiappetta, *supra* note 42, at 108 ("The decision . . . strengthened the view that the Court believed the entire class of computer software related inventions was potentially unpatentable subject matter . . . .").

section 101 in light of a "Method for Updating Alarm Limits" in chemical reactions.[181]

The patent application provided a formula for computing updated alarm limits,[182] but it did not describe the means of setting off the alarm.[183] Consequently, the Court held that the novelty of the invention was only in the "mathematical algorithm or formula" used.[184] Rejecting the "narrow reading of *Benson*," that patentability was based on whether a claim "wholly pre-empts" a mathematical algorithm, the *Flook* Court stated that a mathematical algorithm is patentable when there is an "inventive concept in its application" to a physical process.[185] Since the concept of using alarm limits to monitor chemical reactions was known in the field, the Court held that the application was not inventive, and therefore the invention was not patentable subject matter under section 101.[186] This holding was damaging to software inventions because the claim was for an improved method of calculation used for a specific purpose, and it was still held unpatentable. Although the *Flook* Court left open the possibility of patentable software inventions,[187] most software inventions seemed precluded by this holding.[188]

---

181. *See Flook*, 437 U.S. at 585.

182. In this invention, if the temperature or pressure in the reaction reached a certain limit, the alarm would sound. *See id.*

183. *See id.* at 586.

184. *See id.* at 585–86.

185. *Id.* at 594.

186. *See id.* at 594–95.

187. *See id.* at 584. Justice Stevens wrote:

> To a large extent our conclusion is based on a reasoning derived from opinions written before the modern business of developing programs for computers was conceived. The youth of the industry may explain the complete absence of precedent supporting patentability. Neither the dearth of precedent, nor this decision, should therefore be interpreted as reflecting a judgment that patent protection of certain novel and useful computer programs will not promote the progress of science and the useful arts, or that such protection is undesirable as a matter of policy.

Id. at 584.

188. *See* Claus D. Melarti, State Street Bank & Trust Co. v. Signature Financial Group, Inc.*: Ought the Mathematical Algorithm and Business Method Exceptions Return to Business as Usual?*, 6 J. INTELL. PROP. L. 359, 367 (1999) ("The effective result of the [*Flook*] decision was to extend the *Gottschalk* rule such that not only were mathematical algorithms not patentable, but most any process utilizing such algorithms was invalidated as well.").

3. *Diamond v. Diehr*

In the last and most recent case in the trilogy, *Diamond v. Diehr*,[189] the Supreme Court finally acknowledged that some software inventions are patentable subject matter under section 101.[190] *Diehr* upheld the patentability of a "process for molding raw, uncured synthetic rubber into cured precision products" using the Arrhenius equation,[191] a known chemical formula.[192] By considering certain variables and continuously calculating the time at which the molding presses should turn off, the computer in the patent application could more accurately cure the rubber products.[193] The Court held that software that is linked to a physical structure or process would be patentable subject matter under section 101, even if an algorithm, previously held unpatentable, were involved in the process.[194] The difference between unpatentable and patentable subject matter was whether the claim was seeking to patent the algorithm "in the abstract," or as applied "in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect."[195] Viewed as a whole, the *Diehr* invention looked like a more traditional industrial process than a computer program and was therefore patentable.[196] Notably, however, the Court failed to expressly overrule *Benson* or *Flook*.[197] The three cases are difficult to synthesize into a useful proposition.[198]

---

189. 450 U.S. 175 (1981).

190. *See id.*

191. *Id.* at 177.

192. The Arrhenius equation equates the rate constant of a chemical reaction, k, with the product of a constant, A, representing the molecular orientations during a collision, and the fraction of collisions with sufficient energy to produce a reaction at a given temperature. *See* STEPHEN S. ZUMDAHL, CHEMISTRY 546 (Mary Le Quesne ed., 2d ed. 1989).

193. *See Diehr*, 450 U.S. at 177.

194. *See id.* at 187–88.

195. *Id.* at 175, 192.

196. *See id.* at 184, 188.

197. *See* Chiappetta, *supra* note 42, at 109.

198. *See* Steven L. Friedman et al., State Street Bank and Trust Company v. Signature Financial Group Inc.: *At the Intersection of Technology, Commerce and the Law*, IPL NEWSL. (American Bar Association Section of Intellectual Property Law, Chicago, IL), Spring 1999, at 8, 10, 16 n.7 (stating that "the result and spirit of *Diehr* is irreconcilable with *Benson* and *Flook*").

C. *The Appellate Response to the Supreme Court: A Two-Part Test*

The three Supreme Court cases led the CCPA to develop a two-step test to determine if computer software is patentable subject matter.[199] The Freeman-Walter-Abele test (the "FWA test"), based on three CCPA cases,[200] asks whether an invention is an application of a mathematical algorithm (patentable subject matter) or is merely an abstract mathematical algorithm (unpatentable subject matter).[201] First, the court determines whether the claim directly or indirectly recites an algorithm, as described in *Benson*. A claim that does not recite an algorithm at all and falls within one of the four statutory categories is patentable subject matter under this test.[202] The second determination evaluates whether the claim involves an application of the algorithm to a specific process or physical structure.[203]

The Federal Circuit was inconsistent in deciding cases after developing the FWA test, sometimes finding computer software patentable subject matter and sometimes not.[204] More importantly, the Federal Circuit did not always use the FWA test.[205] One explanation for this erratic behavior is that the Federal Circuit hears cases in three-judge panels and different panels hear different cases.[206] Although the panels are supposed to follow the other panels' decisions, they do not always do so.[207] This inconsistency has led one commentator to note:

---

199. *See* Maximilian R. Peterson, Note, *Now You See It, Now You Don't: Was It a Patentable Machine or an Unpatentable "Algorithm"? On Principle and Expediency in Current Patent Law Doctrines Relating to Computer-Implemented Inventions*, 64 GEO. WASH. L. REV. 90, 105–07 (1995).

200. *See In re* Abele, 684 F.2d 902 (C.C.P.A. 1982); *In re* Walter, 618 F.2d 758 (C.C.P.A. 1980); *In re* Freeman, 573 F.2d 1237 (C.C.P.A. 1978).

201. *See* Arrhythmia Research Technology, Inc. v. Corazonix Corp., 958 F.2d 1053, 1058 (Fed. Cir. 1992).

202. *See id.*

203. *See* Martone, *supra* note 18, at 618.

204. *See supra* notes 23–24 and accompanying text for inconsistent cases.

205. *See In re* Alappat, 33 F.3d 1526 (Fed. Cir. 1994) (en banc) (deciding case without the FWA test). *See generally* Weissman, *supra* note 15, at 535–43.

206. *See* Carl Chan, Note, *The Patentability of Software Data Structures after* Lowry *and* Warmerdam, 32 NEW ENG. L. REV. 899, 912 n.133 (1998).

207. *See id.*

> [S]oftware is [not] per se unpatentable subject matter. Instead, the courts have turned the issue into an examination of the relationship that the software must have with physical elements. It is apparent that the claimed invention must meet a threshold level of physical element interconnected with the program. Yet, *the precise physical elements which are sufficient to render a program patentable have yet to be resolved.*[208]

State Street now has resolved this inconsistency.[209]

V. *STATE STREET*: THE FLOODGATES OPEN

A. *Background of the Case*

Signature Financial Group, Inc. ("Signature"), an administrator and accounting agent of mutual funds, owned United States Patent 5,193,056 (the "'056 patent"), which described a computer program that implements an investment structure developed for use by Signature.[210] The program, known by the proprietary name "Hub and Spoke," facilitates a structure whereby mutual funds—the "Spokes"—pool their assets in an investment portfolio—the "Hub"—organized as a partnership for tax purposes.[211] This investment structure provides the administrator of a mutual fund with the advantage of economies of scale, including reduced overhead and research costs.[212]

The program calculates daily changes in the allocation of assets among the mutual funds invested in the partnership, based on the daily changes in the mutual funds' investments and the percentage share of each fund in the portfolio.[213] The program also tracks the data relevant to calculating aggregate year-end income, expenses, and capital gain or loss for accounting and tax purposes.[214] Government regulations require accurate calculations of these figures to the penny within an

---

208. Weissman, *supra* note 15, at 549 (emphasis added).

209. *See* Ellis & Chatterjee, *supra* note 6 (calling *State Street* "one of the most important decisions relating to the patentability of software.").

210. *See* State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1370 (Fed. Cir. 1998), *cert. denied*, 119 S. Ct. 851 (1999).

211. *See id.* at 1370.

212. *See id.*

213. *See id.* at 1371.

214. *See id.*

hour and a half from the close of the market, because the mutual funds sell shares to the public and the price of these shares is dependent on the mutual funds' percentage interest in the portfolio.[215]

State Street Bank & Trust Co. ("State Street Bank"), the alleged infringer, is a leader in the financial services market, controlling approximately 44 percent of the United States mutual fund market and having revenues of nearly $3.5 billion.[216] The '056 patent provided Signature with the means to challenge State Street Bank's giant market share.[217]

State Street Bank initially attempted to negotiate a license to use the patented software system, but when negotiations broke down, it filed a declaratory judgment action in the United States District Court for the District of Massachusetts to determine the validity of the '056 patent.[218] The District Court granted summary judgment for State Street Bank, finding that Signature's patent was invalid on the basis of section 101, under either the mathematical algorithm or business method exception.[219] Signature appealed to the Federal Circuit.[220]

B. The Federal Circuit Opinion

The Federal Circuit, Judge Rich, writing for the three-judge panel including Judges Plager and Bryson, reversed the trial court's summary judgment ruling.[221] The court restricted the mathematical algorithm exception for section 101 subject matter to those situations in which the mathematical calculation merely constitutes an abstract idea consisting of disembodied concepts without utility.[222] Noting that the language of section 101 is expansive and demonstrates Congress's intent not to place any additional restrictions on patentable subject matter beyond the four statutory categories, the court stated

---

215. *See id.*

216. *See* Ellis & Chatterjee, *supra* note 6.

217. *See id.*

218. *See* State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 927 F. Supp. 502 (D. Mass. 1996).

219. *See id.* at 517.

220. *See* State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1370 (Fed. Cir. 1998), *cert. denied*, 119 S. Ct. 851 (1999).

221. *See id.* at 1377.

222. *See id.* at 1373.

that it is improper to read limitations into section 101.[223] Consequently, the court found that mathematical algorithms would be unpatentable only if they fit within the abstract ideas exception created by the Supreme Court in *Diehr*.[224] So long as the abstract idea is applied to produce a "useful, concrete and tangible result," the software invention is patentable subject matter.[225]

The court considered the CCPA's FWA test in light of the *Diehr* decision[226] and *Diamond v. Chakrabarty*,[227] and held that the FWA test has little applicability to determining the presence of patentable subject matter in software inventions.[228] The court noted that, contrary to the FWA approach, the particular statutory category chosen to claim the subject matter is not of particular relevance and should not be a focus of the court, so long as the claimed invention falls into one of the statutory categories.[229] Rather, the court should focus on the "essential characteristics of the subject matter, in particular, its practical utility."[230] Here, the usefulness of the Hub and Spoke computer program "renders it statutory subject matter, even if the useful result is expressed in numbers, such as price, profit, percentage, costs, or loss."[231]

The court also laid to rest the business methods exception, claiming that neither it nor its predecessor court, the CCPA, had ever relied on the exception as the primary ground for rendering an invention unpatentable.[232] Instead, business methods should be subject to the "same legal requirements for patentability as applied to any other process or method."[233] Concluding that Signature's claims constituted statutory sub-

---

223. *See id.*
224. *See id.*
225. *Id.* at 1375.
226. Diamond v. Diehr, 450 U.S. 175, 185 (1981).
227. 447 U.S. 303, 309 (1980) (stating that human synthesized bacteria were patentable subject matter since "anything under the sun that is made by man" is patentable subject matter (quoting S. Rep. No. 82-1979, at 5 (1952); H.R. Rep. No. 82-1923, at 6 (1952))).
228. *See* State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1373–74 (Fed. Cir. 1998), *cert. denied*, 119 S. Ct. 851 (1999).
229. *See id.*
230. *Id.* at 1375.
231. *Id.*
232. *See id.* ("We take this opportunity to lay this ill-conceived exception to rest.").
233. *Id.*

ject matter, the Federal Circuit reversed the district court's decision.

State Street Bank's appeal to the United States Supreme Court was denied certiorari on January 11, 1999.[234] The news that the decision would stand caused "shock waves" in the various software-related industries, as discussed below.[235]

VI. THE AFTERMATH OF THE FLOOD—CONSEQUENCES OF *STATE STREET*

With the specter of a section 101 rejection essentially eliminated, whole areas of business, such as the finance and insurance sectors, which never had previously considered patents to be viable options for protecting their intellectual property, must now do so.[236] They are now faced with the reality of acquiring patents themselves, or more frighteningly, watching their competitors acquire them first.[237]

As discussed in Part I, certainty in the intellectual property protection scheme is crucial in order to maximize the incentives for innovation. Because *State Street* was such a dramatic departure from prior precedent, it created an uncertainty in the computer-based industries. A dilemma facing computer-related industries is whether to spend the resources needed to acquire patent protection without knowing if the relaxed standard of patentable subject matter in *State Street* will continue under subsequent court consideration.

The Supreme Court helped resolve some of this uncertainty by denying certiorari. While the denying of certiorari by the Supreme Court holds no precedential value,[238] the decisions of the Federal Circuit on patent issues are binding on all lower

---

234. State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 119 S. Ct. 851 (1999).

235. *See* Paul A. Beck, State Street Bank *Case Causes Shock Waves in Banking and Financial Industry*, 147 PITT. LEGAL J. 7, 7 (1999).

236. *See* Melarti, *supra* note 188, at 387–88.

237. Just as the restriction of the mathematical algorithm exception has affected computer-based industries, the elimination of the business method exception has had a huge effect on traditional business areas, as well as more modern computer-related industries. These areas include franchise businesses and the rapidly growing e-commerce industry. *See* W. Scott Petty, State Street Bank *Fuels Boom in Business Model Patents*, INTELL. PROP. TODAY 30 (1999).

238. *See* Brown v. Allen, 344 U.S. 443, 497 (1953) (quoting Sunal v. Large, 332 U.S. 174, 181 (1947)) ("[D]enial of certiorari cannot be interpreted as an 'expression of opinion on the merits.'").

courts unless contradicted by federal statute or a Supreme Court decision. Consequently, the *State Street* decision is now the law of the land unless the decision is revisited.

Because the Federal Circuit hears cases in three-judge panels that do not always follow each other, it is possible that the other panels will simply refuse to follow *State Street*. Again, this is cause for undesirable uncertainty in the law. Fortunately, one of the other Federal Circuit panels has already spoken on the issue.

In *AT&T Corp. v. Excel Communications, Inc.*,[239] the plaintiff owned a patent entitled "Call Message Recording for Telephone Systems."[240] The invention basically added an indicator to the message record for long-distance phone calls, which helped long-distance carriers provide differential billing treatment to subscribers, depending on whether the subscriber called someone with the same or with a different long-distance carrier.[241] AT&T sued Excel for infringement of ten method claims in its patent.

Decided before the Federal Circuit's *State Street* opinion, the *AT&T* district court held the method claims invalid based on section 101, because the claims fell within the mathematical algorithm exception.[242] Consequently, as in *State Street*, the *AT&T* district court granted summary judgment in favor of the defendant.[243] AT&T appealed to the Federal Circuit.[244]

Judge Plager, writing for Judges Clevenger and Rader, reversed the trial court's grant of summary judgment and remanded for further proceedings.[245] Noting that "at the time the trial court made its decision, that court did not have the benefit of this court's explication in *State Street* of the mathematical algorithm issue,"[246] the court further solidified the *State Street* decision by stating that "the judicially-defined proscription against patenting of a 'mathematical algorithm,' to the extent such a proscription still exists, is narrowly limited to mathe-

239. 172 F.3d 1352 (Fed. Cir. 1999), *cert. denied*, 120 S. Ct. 368 (1999).
240. *Id.* at 1353.
241. *See id.*
242. *See id.* at 1355.
243. *See id.* at 1353.
244. *See id.*
245. *See id.*
246. *Id.* at 1358.

matical algorithms in the abstract."[247] In conclusion, the court announced that "the focus is understood to be not on whether there is a mathematical algorithm at work, but on whether the algorithm-containing invention, as a whole, produces a tangible, useful, result."[248]

Given that two different panels have now adopted the reasoning in *State Street*, it is reasonable to believe, although certainly not guaranteed, that the Federal Circuit will continue to follow *State Street*. This expectation provides a large measure of needed security in the area of computer software patents.

After *State Street* and *AT&T*, mathematical algorithms are not patentable subject matter insofar as they are merely abstract ideas.[249] Furthermore, the business method exception is an "ill conceived exception" that is laid to rest.[250] Neither of these alternate exceptions is found in section 101 of the Patent Act. In addition, neither exception advances the constitutional mandate of promoting progress in the useful arts, since they have been used to keep otherwise potentially patentable inventions from being fully disclosed to the public through the patenting process. Given these rationales and the strong congressional intent to allow "anything under the sun that is made by man" to be patentable subject matter,[251] both cases were correctly decided.

One immediate consequence of *State Street*, however, has been a "'boom' in business-method-related [software] patent applications."[252] This trend will most likely continue due to the support shown in *AT&T*. Furthermore, the growing use of the internet by the world of business and finance will only augment this trend.[253] Unfortunately, the PTO is not prepared to deal with the influx of new applications.[254]

---

247. *Id.* at 1356.

248. *Id.* at 1361.

249. *Cf.* Diamond v. Diehr, 450 U.S. 175 *passim* (1981).

250. *See* State Street Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1375 (Fed. Cir. 1998), *cert. denied*, 119 S. Ct. 851 (1999).

251. Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980) (quoting S. Rep. No. 82-1979, at 5 (1952); H.R. Rep. No. 82-1923, at 6 (1952)).

252. Pezzano, *supra* note 5, at 2 (noting that the decision has "triggered a race in the e-commerce/financial industries to obtain patent protection on computer software-related methods of doing business/finance").

253. *See id.*

254. *See* James Pooley & Colleen Pouliot, *Defensive Strategies in Software Patent Litigation*, 17 ACAA DOCKET 34 (1999).

To its credit, the PTO has hired twenty new patent examiners to review electronic commerce patents in response to the *State Street Bank* decision.[255] However, more must be done. In addition to hiring more examiners, these examiners need to have backgrounds in computer science. Until recently, the PTO did not recognize experience in that field as a valid qualification for becoming a patent attorney.[256]

Once enough qualified examiners are available to handle computer software applications, their inquiry needs to focus on the overall patentability question. *State Street* and *AT&T* hold that section 101 is not meant to be a significant barrier to patentability of computer software inventions. The remaining requirements of disclosure, novelty, and nonobviousness need to be enforced uniformly and strictly. One of the consequences of the insufficient resources at the PTO has been a large number of issued—or soon-to-be issued—patents that should never have been allowed due to failure to meet the novelty or nonobviousness requirements.[257]

Even if the proper resources are available and the proper inquiries are made, however, there is still a significant problem with the current patent system's application to computer software inventions. One significant difference between computer software technology and other types of technology is the degree to which the advances are made available to the public in an understandable printed form (including patents).[258] In most technical areas, there is a strong history of rapid publishing and patenting of new inventions in order to begin the period of protection against competitors as quickly as possible.[259] Software inventions, however, historically have been maintained as in-house trade secrets due to the reluctance of the PTO to grant software patents that would withstand judicial scrutiny. Even today, as more software patents are issued and upheld, mass-produced software is often released in object code only, which

---

255. *See* Pezzano, *supra* note 5, at 2. In addition, the PTO plans to double the patent examiner corps in the near future. *See Patent Answers*, GOV'T EXECUTIVE, Feb. 1999, at 81, 82.

256. *See* Pooley & Pouliot, *supra* note 254, at 34.

257. *See id.*

258. *See id.*

259. The Patent Act, while not a pure "first to file" system, does give certain procedural preference to the first application filed on a particular invention. *See* 35 U.S.C. § 135 (1994).

tends to be incomprehensible to the average human reader. Both of these trends have kept software advances effectively out of the public domain and therefore out of the prior art available to the patent examiners when they make their patentability determinations.

The absence of software advances in the public domain has several negative effects. One is that software companies often spend time, effort, and money reinventing the wheel since they are not aware of—or perhaps do not want to pay a licensing fee for—existing technology.[260] More importantly, the patent examiners at the PTO are unable to make a thorough and accurate search of the prior art for novelty or nonobvious determinations.[261] Allowing computer software to be patentable subject matter without the limitations imposed by the novelty and nonobvious requirements means that invalid patents will be issued.[262] This will undermine confidence in the patent system's ability to handle computer software and will likely lead to an increase in expensive litigation due to the many issued "invalid" patents. The problems stated above need to be resolved in order to satisfy the constitutional mandate that the patent system "promote the Progress of Science and the useful Arts."[263]

VII. A PLAN FOR THE FUTURE

Courts have struggled with extending patent protection to computer programs because a set of instructions to run computer hardware is by definition an algorithm.[264] Algorithms can be thought of as abstract ideas or laws of nature, and to that end they are not patentable subject matter under section 101.[265] However, concluding that computer software "is inherently unpatentable would effectively exclude many aspects of utilitarian computer programs from both copyright protection (as methods of operation) and from patent protection (as non-

260. *See* Bender, *supra* note 24, at 160.
261. *See id.*
262. *See id.*
263. U.S. CONST. art. 1, § 8, cl. 8.
264. *See* Gottschalk v. Benson, 409 U.S. 63, 65 (1972) ("A procedure for solving a given type of mathematical problem is known as an 'algorithm.'").
265. *See id.*

statutory subject matter)."[266] Consequently, patent lawyers and courts have variously described computer software as a process, a machine, or an article of manufacture by stretching the definitions and manipulating the claim language.[267] These efforts have complicated the patents so that the public cannot understand them and have frustrated efforts to accurately search and classify software inventions.[268] *State Street* has provided a solution to this problem in that computer software is patentable subject matter so long as it is applied to produce a useful, concrete, and tangible result.

Despite the fact that it appears likely that the Federal Circuit will follow *State Street*, it is still possible that the Supreme Court will revisit this issue and decide to take another path. In order to promote certainty in the field, the Patent Act itself should be amended to codify the holding of *State Street*. The following is one suggestion:

> § **101. Inventions patentable (proposed)**: Whoever invents or discovers any new or 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. *To the extent that a computer-implemented program or piece of software is a practical application of an abstract idea, a natural phenomenon, or a law of nature, it is also patentable subject matter under this section, subject to the conditions and requirements of this title.*

This amendment would have the advantage of consistency with the prior Supreme Court cases on the issue and would remove any remaining uncertainty on the issue of whether computer software is by its very nature patentable subject matter.

More patent examiners with computer software experience need to be provided by any means necessary. Perhaps the funding could come from an increase in patent application fees. This has the advantage of providing a simple method of raising revenue. However, it has the disadvantage of potentially ex-

---

266. Martone, *supra* note 18, at 615.

267. *See id.*

268. *See* Lee A. Hollaar, *Justice Douglas Was Right: The Need for Congressional Action on Software Patents*, 24 AIPLA Q.J. 283, 288 (1996).

cluding some inventors from the patent process due to the increased expense.

One possible solution to the lack of available prior art discussed in Part VI is to establish national databases that catalog software inventions.[269] Apple Computer, IBM, the University of Michigan, and other computer companies founded one such database, the nonprofit Software Patent Institute, in 1992.[270] Such efforts should help immensely in facilitating the difficult task of researching prior art.

Like computer software, semiconductor chips have industrial designs that are "rarely inventive," and the ease of copying them undermines inventors' ability to recoup research and development costs.[271] Congress attempted to correct this problem in the semiconductor chip industry by passing the Semiconductor Chip Protection Act of 1984.[272] It has been argued that it may also need to protect the computer software industry for the same reasons.[273] A significant advantage of congressional intervention would be an incorporation of all existing forms of intellectual property protection, resulting in a system that adequately protects software inventions without substantially disrupting existing intellectual property law.[274] This, however, is a monumental task and even Congress may be unable to tackle it.

Short of a complete congressional overhaul of the intellectual property protection afforded computer software,[275] the courts should look to *State Street* for the proposition that computer software is patentable subject matter under section 101, so long as it is not pure thought. This means that section 101 will be an easy hurdle to pass, in accordance with Congress's

---

269. *See* Bender, *supra* note 24, at 160–61.
270. *See id.*
271. *See* Samuelson et al., *supra* note 49, at 2346.
272. *See* 17 U.S.C. §§ 901–914 (1994).
273. *See* Samuelson et al., *supra* note 49, at 2346.
274. *See* Hollaar, *supra* note 268, at 284.
275. *See* Gottschalk v. Benson, 409 U.S. 63, 73 (1972).

> If these problems are to be patentable, considerable problems are raised which only committees of Congress can manage, for broad powers of investigation are needed, including hearings which canvas the wide variety of views which those operating in this field entertain. The technological problems tendered in the many briefs before us indicate to us that considered action by the Congress is needed.

Id. (footnotes omitted).

determination that "anything under the sun that is made by man" is patentable subject matter.[276] The resources of the PTO are better spent searching the prior art to make novelty, nonobviousness, and full disclosure determinations than arguing about algorithms.[277] The floodgates will be opened and the PTO will be swamped with new patent applications, but the time saved by the resolution of the section 101 question should allow patent attorneys to prepare, and the PTO to issue, patents more efficiently. More importantly, the software industry that has become such a large part of our modern high-tech workplace will have the intellectual property protection that it requires.

The proposals explained above should decrease the time it takes for a patent to issue, since the subject matter inquiry will be greatly simplified. Not only will this be beneficial to the applicant, but it will also serve the constitutional objectives of promoting the progress of the useful arts by getting protected software inventions into the public domain in less time.

CONCLUSION

The *State Street* decision is a step in the right direction. The existing intellectual property protection schemes have not kept up with technological advances in fields such as computer software. Although computer software historically has received copyright and trade secret protection, patent protection best serves the needs of software inventions in that the functionality of the program is protected.

The PTO and the courts, however, historically have been reluctant to grant such patent protection due to a conception of computer software as a mathematical algorithm classified as a judicial exception to section 101 of the Patent Act. *State Street* changes this conception and allows much-needed patent protection for computer software.

The focus for computer software patents should be on the exact scope of the patent, not on whether or not the invention is patentable subject matter. This comment proposes an amendment to section 101 of the Patent Act so that *State Street* is ex-

276. Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980) (quoting S. REP. NO. 82-1979, at 5 (1952); H.R. REP. NO. 82-1923, at 6 (1952)).

277. *See* Hollaar, *supra* note 268, at 289.

pressly codified. Such an amendment would allow the PTO to focus on other patentability tests to determine whether or not a particular application should issue. As long as the PTO has the resources to properly examine both the patent application and relevant prior art, the patents that ultimately do issue should continue to further the constitutional mandate of progress in the useful arts.