UIC John Marshall Journal of Information Technology & Privacy Law

Volume 23
Issue 4 *Journal of Computer & Information Law - Summer 2005*

Article 4

Summer 2005

Software Patents on Both Sides of the Atlantic, 23 J. Marshall J. Computer & Info. L. 815 (2005)

Jack George Abid

SOFTWARE PATENTS ON BOTH SIDES OF THE ATLANTIC

JACK GEORGE ABID†

I. INTRODUCTION

Numbers are always a good starting point. One good number is one hundred fifty thousand; the approximate number of issued software patents in the United States ("U.S."). [1] Another sizable number is twenty thousand; this is the approximate number of software patents issued by the European Patent Office ("EPO"). [2] Given these numbers, the average observer could easily conclude that software is patentable in both the U.S. and the European Union ("EU"). In the case of the U.S., the numbers bear out the effects of a string of court decisions that eroded the prior proscription of software patentability. The trend exhibited by the EPO is the result of a string of similar decisions by the Technical Board of Appeal ("Board") that arguably deviate from substantive patent law statutes. [3]

Software patentability is a significant issue at both the corporate and individual levels. The U.S. accounts for nearly one half of the software industry's worldwide revenues of one hundred seventy billion

† Jack George Abid, Juris Doctor candidate May 2006 University of Florida Fredric G. Levin College of Law; Bachelor of Science in Electrical Engineering, Georgia Institute of Technology 2003; and Bachelor of Science in Physics, Jacksonville University 2003. I wish to thank my family and friends for their patience and support. I wish to express my gratitude to my parents, Joseph and Sylvia; I would not be the person I am today without their constant support and love.

1. Robert McMillan, *Torvalds Joins in Anti-Patent Attack*, http://www.techworld.com/applications/news/index.cfm?NewsID=3059 (accessed Nov. 16, 2005).

2. Foundation for a Free Information Infrastructure ("FFII"), *European Software Patent Statistics*, http://swpat.ffii.org/patents/stats/index.en.html#jarappl (accessed Mar. 4, 2005). The number of existing software patents issued by the United States Patent and Trademark Office ("PTO") and the European Patent Office ("EPO") is debatable since there is no clear definition of a pure software patent.

3. Convention on the Grant of European Patents, Oct. 5, 1973, art. 52(4), 1065 U.N.T.S. 199 ("EPC"), http://www.european-patent-office.org/legal/epc/e/ma1.html#CVN (accessed Mar. 31, 2005).

dollars.[4] The U.S. has also taken the most liberal stance on software patentability.[5] This has many non-U.S. software firms clamoring for equivalent protection in the EU. On the personal level, the patentability of software affects the freedom of expression for every freelance and commercial programmer. Consequently, the debate on software patentability has involved the likes of the Microsoft Corporation and hobbyist programmers.

Notwithstanding the numbers, software patentability has determined opponents on both sides of the Atlantic and has drawn critical commentary from some of society's greatest minds.[6] Fueled by a slew of dubious software patents issued by the U.S. Patent and Trademark Office ("PTO") and the EPO, these opponents argue that software patentability will: result in the issuance of more dubious patents, become an impediment to innovation by destroying small software firms, be the wrong form of intellectual property for software, and cause significant damage to the open source community. These alleged problems with software patentability lead Linus Torvalds to recently declare that "[s]oftware patents are clearly a problem . . . ."[7] One decade earlier, Bill Gates argued that software patentability would have been an impediment to innovation and growth in the software industry during the 1980s.[8] Tim O'Reilly declared software patents as "one more example of

4. All monetary amounts within this Comment are listed in U.S. Dollars unless otherwise noted.

5. Software & Information Industry Association, *Software Industry Statistics for the Third Quarter 2004*, http://www.siia.net/software/pubs/statpage_Q304.pdf (accessed Nov. 2, 2005).

6. The following is a non-exhaustive list of organizations that have adopted positions against software patentability: League for Programming Freedom, *Homepage*, http://lpf.ai.mit.edu/ (accessed Nov. 2, 2005); Foundation for a Free Information Infrastructure, *Homepage*, http://www.ffii.org/ (accessed Nov. 2, 2005); Confédération Européenne des Associations de Petites et Moyennes Enterprises, *Homepage*, http://www.ceapme.org/ (accessed Nov. 16, 2005); SWM Software-Marketing GmbH, *Homepage*, http://www.no softwarepatents.com/ (accessed Nov. 16, 2005). The first four results from a Google search on the World Wide Web for "software patents" are anti-software patent organizations. *See* Google Corp., *Google Search: Software Patents*, http://www.google.com/search?hl=en&lr=&q=software+patents+&btnG=Search (accessed Mar.26, 2005).

7. McMillan, *supra* n. 1. Linus Torvalds is wildly credited with the creation of the Linux Operating System. For more background material on Linus Torvalds, see Gary Rivlin, *Leader of the Free World: How Linus Torvalds Became Benevolent Dictator of Planet Linux, the Biggest Collaborative Project in History*, Wired, http://www.wired.com/wired/archive/11.11/linus.html (Nov. 2003).

8. Memorandum from William H. Gates III Chairman and Chief Software Architect of Microsoft Corp. to Microsoft Employees on Challenges and Strategy, to Microsoft Employees, http://www.bralyn.net/etext/literature/bill.gates/challenges-strategy.txt (May 16, 1991). In an intriguing change of position, the Microsoft Corporation has applied and received several thousand software patents on both sides of the Atlantic. FFII, *Microsoft and Patents*, http://swpat.ffii.org/gasnu/microsoft/index.en.html (Aug. 16, 2004).

an 'intellectual property' milieu gone mad."[9] Software patents are not without their proponents, although seemingly less numerous and less vocal, who argue that software patents are needed to effectively protect the intellectual property of software.

In early 2005, the EU was attempting to codify the unofficial stance on software patentability that the EPO Board has taken.[10] In the furor over proposed software patentability in the EU, troubling accusations have surfaced alleging that large software firms are employing questionable tactics in favor of the "Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions."[11] Software patentability opponents in Europe have likened the EU to a "banana republic" in its behavior in regards to the Software Patent Directive.[12] The debate over the Software Patent Directive has drawn the attention of European and U.S. media alike.[13] It is also being closely watched by U.S. law makers as a precursor to anticipated patent reform in the U.S.

This Comment attempts to address and analyze the current state of software patentability in the U.S. and the EU. Section II gives a brief introduction on the policies that drive changes and developments in patent law, the technical background of software,[14] and the non-patent intellectual property protection for software. Section III discusses the judicial precedent and the current situation surrounding software patents in the U.S. Section IV similarly discusses the legal situation in the EU including proposed changes in the Software Patent Directive. Sec-

---

9. Email from Tim O'Reilly, Founder and CEO, O'Reilly and Associates, to Richard Caley, *O'Reilly Network: Open Source, Patents and O'Reilly*, http://www.oreilly.com/pub/a/oreilly/ask_tim/ 2000/amazon_patent.html (Feb. 28, 2000). O'Reilly and Associates is widely considered to be the best publisher of computer and technology related books in the world.

10. Commission Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer Implemented Inventions, Eur. Parl. Doc. (COM 92 final) 3 ("Software Patent Directive"), art. 3, http://europa.eu.int/comm/internal_market/en/indprop/comp/com02-92en.pdf (accessed Mar. 19, 2005).

11. *Bill Gates threatens to close Navision in Denmark*, The Copenhagen Post, http://www.cphpost.dk/get/85881.html (Feb. 15, 2005).

12. FFII, *Bananas to the German Federal Ministry of Justice: On Software Patents, Consumer Protection, and the European Banana Union*, http://wiki.ffii.org/?BmjBananen Pm050225En (Feb. 25, 2005).

13. *See* Paul Meller, *Plan for Patenting Software Stalls in Europe's Parliament*, N.Y. Times, Feb. 18, 2005, at C13; BBC, *EU Software Patent Law Delayed*, http://news.bbc.co.uk/1/hi/technology/ 4116231.stm (Dec. 22, 2004); Bill Thompson, *Be Careful How You Code*, http://news.bbc.co.uk/1/hi/technology/4104841.stm (Dec. 17, 2004); Bill Thompson, *Fight for the Right to Program* (June 7, 2004), http://news.bbc.co.uk/1/hi/technology/3782771.stm (accessed Nov. 16, 2005).

14. No discussion on the merits of software patentability is feasible without a basic understanding of software and the policies behind patent law.

tion V comparatively analyzes software patentability in the U.S. and EU. Section VI discusses common criticisms of current U.S. policies on software patentability by exposing the problems created by these policies and with suggestions for corrective policies in the future.

II. BACKGROUND

A. Policy Concerns in Patent Law

Every piece of legislation purports to serve a legitimate objective of the state and patent law legislation is no different. The patent system serves the public policy of providing incentive to invent, stimulating investment capital within industry, encouraging public disclosure of new technology, and promoting the beneficial exchange of products, services, and technological information.[15] Patent law legislation facilitates these goals by granting the inventor certain rights that prohibit others from making, using, importing, or selling the patented invention.[16] The patent system combats under-investment in research, development, and related activities.[17] The patent system represents a carefully crafted bargain between society and the inventor, where the inventor exchanges his novel and useful improvement for the right to use and profit from it exclusively for a set period of time.[18] There must be a balance between the cost and the benefit to society and inventor when granting patent rights.[19] If the balance sways too favorably toward society by weakening patent rights, the incentive to innovate will suffer and technological advancement within industry will decline. Conversely, if the patent rights are strengthened excessively, competition will dwindle, and the industry will suffer economic inefficiency.

B. Technical Discussion on Software

Similar to the text of this Comment, software at the code level[20] consists of text characters that signify some sort of instruction for the digital computer. Software is the representation of human ideas in the digital world and is the medium of expression for programmers. Whereas humans communicate with words and sentences, digital computers communicate in a language composed solely in a binary number system con-

15. 60 Am.Jur.2d, *Patents* § 2 (2004).

16. 35 U.S.C.S. § 154 (2004).

17. Erich Kaufer, *The Economics of the Patent System* 22 (Harwood Academic Publishers 1989).

18. *Pfaff v. Wells Elect., Inc.*, 525 U.S. 55, 63 (1998).

19. *Id.*

20. Software at the code level is commonly known as programs, script, code, or source code. Webopedia, *Source Code*, http://www.webopedia.com/TERM/s/source_code.html (accessed Nov. 16, 2005).

sisting of zeros and ones.[21] Since humans think and work best with words and sentences, there must be a translation between the ideas of the programmer to the digital computer medium. A tri-level programming language paradigm was developed to manage this translation from human ideas to digital computer instructions. The three levels of programming languages are high level source code ("source code"),[22] assembly code, and low level machine code ("machine code").[23]

When a software engineer or programmer creates a program, the program or source code is composed of words and symbols that are relatively easy to learn and understand. This is called source code and is commonly how programmers formulate their ideas in the digital world; source code is the programmer's digital canvas. Since source code is readily understandable, access to the source code of any program provides a digital road map of the logic and processes that make up the program.[24] Since this type of programming language is the bread and butter of programmers, software patents typically revolve around source code. Machine code is the language digital computers process directly where each line of code represents a single step instruction for the digital computer to perform. Machine code consists purely of ones and zeros and is generally exceedingly difficult for humans to decipher. Machine code is the language in which the program is traditionally distributed as an end product. Given this esoteric nature of machine code, access to the

21. Humans commonly use what is called a base ten decimal number system consisting of the numerals 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Computers process information in the binary number system which consists solely of the numerals 0 and 1. This binary basis is due to the binary nature of the fundamental component of all computer systems: the transistor. For Example: The binary number 1011 can be represented as a decimal number by $(1 \times 23) + (0 \times 22) + (1 \times 21) + (1 \times 20)$. This simplifies to $(1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1) = 8 + 0 + 2 + 1 = 11$. This means that the binary number 1011 is equivalent to 11 in the decimal number system. Interactive Mathematics Dictionary, *Binary System*, http://www.intermath-uga.gatech.edu/dictnary/descript.asp?termID=54 (accessed Mar. 31, 2005).

22. Examples of source code include Visual Basic, C, C++, Java, Fortran, COBOL. Data structures and subroutines are used to represent logic of the program that is converted to low-level code before processed by the machine. Webopedia, *Program*, http://www.webopedia.com/ TERM/P/program.html (accessed Nov. 16, 2005).

23. Webopedia, *Programming Language*, http://www.webopedia.com/TERM/p/ programming_language.html (accessed Mar. 12, 2005). The graph, *id.*, depicts the three levels of software abstraction with the bottom representing the least abstraction and the top representing the most. The abstraction process is how esoteric machine code is translated to easily understood source code. Humans think better in a more abstracted context; consequently, high level programming languages are the choice of most programmers. *See* Webopedia, *Program*, http://www.webopedia.com/TERM/P/program.html (accessed Nov. 16, 2005).

24. This process of discovering the inner logic and workings of software is commonly known as reverse engineering. Wikipedia, *Reverse engineering of software*, http://en.wikipedia.org/wiki/ Reverse_engineering#Reverse_engineering_of_software (accessed Nov. 16, 2005).

machine code typically does not allow examination of the logic and processes of the program. Assembly code, the third type of programming language, is typically considered a bridge between the source code and the machine code. Assembly code consists of alphanumeric labels that represent basic single step instructions to the computer. Although difficult, a person can interpret what the assembly code means and represents; nonetheless, the extraction of the overall logic and processes of the entire program is difficult.[25]

C. Non-Patent Intellectual Property Protection for Programs

In the early days of the software industry, software firms principally relied on trade secret and unfair competition law to protect their programs.[26] This was an effective way to protect the source code given that the distributed machine code was difficult to reverse engineer into source code.[27] Copyright protection in the U.S. for the source code of software became available first in 1964 when the Register of Copyrights began accepting registrations for source code.[28] Few software firms took advantage of the new form of intellectual property protection and instead chose to maintain their source code as a trade secret since the two forms of protection were mutually exclusive.[29] It was not until the U.S. Congress amended copyright statutes in 1980 that computer software was explicitly recognized as proper subject matter under the U.S. copyright regime.[30] With the relic of registration abrogated, software firms were now presented two avenues of protection for their software: copyright protection and trade secret protection.

European uniformity on copyright protection for software was first attempted by the EU in 1991 with the passage of EU Software Direc-

25. For a more in-depth explanation of the varying levels of computer programming language, *see* Microsoft Encarta Online Encyclopedia 2005, *Computer*, http://encarta.msn.com/encyclopedia_761563087/Computer.html (accessed Mar. 7, 2005).

26. Julie E. Cohen, et al., *Copyright in a Global Information Economy* 240 (Erwin Chemerinsky et al. eds., Aspen Publishers 2002).

27. *Id.*

28. *Id.* In 1964, the prevailing copyright law in the U.S. was federal and was contained within the Copyright Act of 1909. Copyright Act of 1909, ch. 320, §§ 1-64, 35 Stat. 1075 (1909) (current version at 17 U.S.C. §§ 101-1332 (2005)). The copyright regime established by the Copyright Act of 1909 required adherence to certain formalities to maintain copyright protection. *Id.* § 10. One of the formal requirements contained in the Copyright Act of 1909 was that authors had to register their works with the Register of Copyrights before protection attached. *Id.* This registration requirement made copyright protection exclusive of trade secret protections since registration would constitute public disclosure.

29. Cohen, *supra* n. 26, at 240.

30. *Id. at* 242.

tive.[31] At that time, EU member states afforded varying degrees of copyright protection to software with some nations providing none.[32] The European software industry was fledging at the time and was comprised primarily of U.S.-owned firms.[33] The EU Software Directive was intended to put EU software firms on equal footing with their U.S. counterparts.[34] Later, uniformity between the U.S. and the EU was achieved when software copyright protection was granted by the Agreement on Trade Related Aspects of Intellectual Property of 1994 ("TRIPS Agreement"), which defined software as literature entitled to copyright.[35] The U.S. and all EU member states are adherents to the TRIPS Agreement.[36] Both the TRIPS Agreement and the U.S. copyright regimes protect software in source code form, assembly language form, and machine code form.[37]

---

31. Council Directive 91/250 of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J. (L 122).

32. Cohen, *supra* n. 26, at 271.

33. *Id.*

34. Pamela Samuelson, *Intellectual Property Arbitrage: How Foreign Rules Can Affect Domestic Protections*, 71 U. Chi. L. Rev. 223, 239 n.9 (2004).

35. Agreement on Trade-Related Aspects of Intellectual Property Rights, Apr. 15, 1994, Marrakesh Agreement Establishing the World Trade Organization ("TRIPS Agreement"), 1869 U.N.T.S. 299, 33 I.L.M. 1197, http://www.wto.org/english/docs_e/legal_e/27-trips.pdf (accessed Nov. 16, 2005).

> Computer Programs and Compilations of Data
>
> 1. Computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention (1971).
>
> 2. Compilations of data or other material, whether in machine readable or other form, which by reason of the selection or arrangement of their contents constitute intellectual creations shall be protected as such. Such protection, which shall not extend to the data or material itself, shall be without prejudice to any copyright subsisting in the data or material itself.

Id. The U.S. has implemented the WTO agreements. Uruguay Round Agreements Act of 1994, Pub. L. No. 103-465, 108 Stat. 4809. The TRIPS Agreement required parties to adopt the substantive portions of the Berne Convention protecting software. WTO, *Legal texts: The WTO agreements: A Summary of the Final Act of the Uruguay Round*, http://www.wto.org/english/docs_e/legal_e/ ursum_e.htm#nAgreement (accessed Mar. 12, 2005). All WTO members implemented the TRIPS Agreement; WTO membership is enjoyed by the following EU Nations: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, Netherlands, Portugal, Spain, Sweden, and United Kingdom. WTO, *Member Information:*

The European Communities and the WTO, http://www.wto.org/english/thewto_e/countries_e/ european_communities_e.htm (accessed Mar. 12, 2005).

36. WTO, *Understanding the WTO: The Organization Members and Observers*, http://www.wto.org/english/thewto_e/whatis_e/tif_e/org6_e.htm (accessed Mar. 12, 2005). The U.S. and the EU member states are members of the WTO and signatories to the TRIPS Agreement. *Id.*

37. Cohen, *supra* n. 26, at 240; *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983).

With copyright protection for software well established, software firms, especially in the U.S., began to seek broader intellectual property protection for their software by filing copyright infringement suits alleging non-literal infringement against competing firms using questionable tactics.[38] These arguments in favor of expanding copyright protection led some courts to postulate them as seeking patent-like protection and that patents may be better suited at protecting software.[39] Simultaneously, the issue of software patentability arose when the PTO and the EPO began issuing patents on software implemented inventions.[40]

III. THE LAW OF THE U.S.

The patent legislation of the U.S. is contained entirely within Title 35 of the United States Code.[41] The patent legislation is passed under the constitutional power "[t]o 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."[42] In order for an invention to be patentable under the Patent Act, the invention must be useful,[43] novel,[44] and non-obvious.[45] Novelty is determined by comparing the subject invention to the prior art.[46] The non-obviousness requirement is satisfied when the invention's improvements upon the prior art are considered non-obvious to a person of ordinary skill in the pertinent art.[47] In addition to being useful, novel, and non-obvious, the invention must fit into one of the four categories of proper subject matter

38. *Computer Assoc. Intl., Inc. v. Altai, Inc.*, 982 F.2d 693 (2d. Cir. 1992) (noting that Defendant's software performed the same operations as the copyright holder's but implemented it with completely unique source code); *Lotus Dev. Corp. v. Borland Intl., Inc.*, 49 F.3d 807, 810 (1st Cir. 1995). *aff'd*, 516 U.S. 233 (1996) (noting Defendant copied the menu tree of the competing program with wholly unique source code).

39. *Computer Associates, Inc.*, 982 F.2d at 712; *Lotus*, 49 F.3d at 819 (Boudin, J., concurring).

40. FFII, *supra* n. 2.

41. Patent Act, 35 U.S.C. §§ 1-376 (2005).

42. U.S. Const. art. I, § 8, cl. 8.

43. 35 U.S.C. § 101 (2005).

44. *Id.*

45. 35 U.S.C. § 103 (2005).

46. 35 U.S.C. § 102 (2005); Black's Law Dictionary (Bryan A. Gardner Ed., 8th ed., West 2004).

Prior art [is the k]nowledge that is publicly known, used by others, or available on the date of invention to a person of ordinary skill in an art, including what would be obvious from that knowledge. Prior art includes (1) information in applications for previously patented inventions; (2) information that was published more than one year before a patent application is filed; and (3) information in other patent applications and inventor's certificates filed more than a year before the application is filed.

Id.

47. 35 U.S.C. § 103(a) (2005).

set out in the statute. Congress laid out the categories for patentable subject matter as any "process, machine, manufacture, or composition of matter" that satisfied the previous statutory requirements for patentability.[48] The U.S. Congress in passing the Patent Act left the exact boundaries of the classifications to courts. Determination of proper patentable subject matter is a job the courts have long performed.[49] Over the years, the courts in the U.S. carved out exceptions to patentable subject matter: laws of nature and mathematical formulas,[50] natural phenomena,[51] abstract ideas,[52] and patents for the machine where the result is claimed.[53]

The U.S. courts have approached expansions in patentable subject matter cautiously, especially in regards to inventions that approach the prohibited subject matter of abstract ideas.[54] The U.S. courts traditionally required some physical change or effect to remove an invention from unpatentable abstract idea. The Court, in *Gottschalk v. Benson*, initially threw software in with unpatentable mathematical formula.[55] The subject invention was a mathematical algorithm developed to convert decimal numbers into binary numbers.[56] The algorithm had no application outside of the digital computer world; thus, this invention was a roundabout attempt at patenting software.[57] The Court ruled that the issuance of a patent on this invention would grant the patentee a prohibited monopoly on a mathematical formula.[58] Consequently, the Court ruled that the numerical conversion software was not patentable.[59] In ruling against patentability, the Court indicated that the earmarks of patentability for process claims that do not involve some particular machine was the transformation or reduction to another state or thing. This represented the Court's aversion to patentability as the subject invention ap-

48. 35 U.S.C. § 101 (2005).

49. *Dolbear v. American Bell Tel. Co.*, 126 U.S. 1, 13 (1888) (landmark decision where the Supreme Court of the U.S. found that "[t]he method of and apparatus for transmitting vocal or other sounds telegraphically [telephone] . . . by causing electrical undulations, similar in form to the vibrations of the air accompanying the said vocal or other sounds" to be proper patentable subject matter) *Id.* at 531.

50. *Mackay Radio & Tel. Co. v. Radio Corp. of America*, 306 U.S. 86, 94 (1939).

51. *Funk Bros. Seed Co. v. Kalo Inoculant Co.*, 333 U.S. 127, 130 (1948).

52. *Le Roy v. Tatham*, 55 U.S. 156 (1852). "A principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right. Nor can an exclusive right exist to a new power, should one be discovered in addition to those already known." *Id.* at 175.

53. *Expanded Metal Co. v. Bradford*, 214 U.S. 366, 385 (1909).

54. *See Halliburton Oil Well Cementing Co. v. Walker*, 329 U.S. 1 (1946).

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

56. *Id.* at 66-67; *see supra* n. 21.

57. *Gottschalk*, 409 U.S. at 64.

58. *Id.* at 71-72.

59. *Id.* at 73.

proached the abstract idea barrier.[60] The Court expressly stated that an intangible invention would not satisfy statutory muster; the invention must have some physical effect to be patentable.[61]

The turning point decision for software patentability was *Diamond v. Diehr*, which was handed down by the Supreme Court of the U.S. in 1981.[62] The inventors in *Diehr* were challenging the invalidation of their process claims for the curing and molding of raw rubber into precision finished products.[63] The subject invention was the use of digital computer in conjunction with the normal curing process to improve the final product.[64] Although the invention improved the state of the art, the invention fundamentally was the use of a digital computer programmed with basic mathematical formulas to improve on an established physical process.[65] Under the rules of claim interpretation promulgated by the *Gottschalk* decision, the invention was not patentable subject matter.[66]

Instead of invalidating the process patent because it used a mathematical formula and digital computer, the Court chose to take another route and examined the invention with a broader perspective.[67] The Court analyzed the process patent as a whole, without looking at the old or new elements individually, in order to be fair to the process application since many new processes combine old and new elements.[68] The inventors were not seeking a patent on the mathematical formula itself but for its use in the process to cure rubber.[69] The Court held that the use of a mathematical formula or digital computer for one step of the process, that was otherwise patentable process or apparatus, did not

60. *Id.* at 70.

61. *Id.*

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

63. *Id.* at 177.

64. *Id.* at 177-81. The process of curing rubber involves placing the mixing agents with uncured polymers into a heat press. *Id.* at 177, n. 1. The press ostensibly would be in the shape desired for the end rubber product. The raw rubber is then cured for a set time period, which is determined through the use of well established mathematical formulas, derived from the temperature of the press and thickness of the mold. *Id.* at 177. The state of art in the rubber industry did not allow for the accurate cures because the temperature of the molding press could not be precisely measured throughout. *Diehr*, 450 U.S. at 178. This problem made determination of accurate mold time fairly difficult.

The inventors in *Diehr* did not bring anything fundamentally new to the process of curing rubber; they still used the standard mixture and molding press as the prior art did. The use of a digital computer to constantly measure the press time and calculate, using the established mathematical formulas, the mold press duration in real time was the significant improvement. This allowed for accurate cure times and for a properly cured finished rubber product. *Id.* at 178-79.

65. *Id.* at 179-180.

66. *Gottschalk*, 409 U.S. at 64.

67. *Diehr*, 450 U.S. at 187.

68. *Id.* at 188.

69. *Id.* at 187.

render the entire process unpatentable by the use of that formula or computer.[70] In moderating their holding, the Court stated that when a patent incorporates a mathematical formula or scientific principle, there must be an inquiry into whether the patent is essentially seeking claim on the "formula in the abstract."[71]

After the *Diehr* decision,[72] software related patents began to trickle into the PTO as skillful patent prosecutors began to couch their software claims into patentable processes; thus the once unpatentable software claim was made patentable simply by reciting certain "magic words."[73] Subsequent to *Diehr*, to properly parse the patentable from the unpatentable, the courts developed the *Freeman-Walter-Abele* test[74] to implement the *Diehr* directed inquiry into whether the patent is seeking claim on the mathematical formula in abstract.[75] This prosecution technique of couching unpatentable software claims within patentable process claims was the standard practice until the Federal Circuit handed down *In re Alappat*.[76] The Federal Circuit expanded the patentability of software by holding that drafting claims that include a necessary general purpose digital computer satisfied the *Diehr* inquiry into otherwise patentable subject matter.[77] The court reasoned that a general purpose computer was converted, for patentability reasons, into a special purpose computer once it operates pursuant to some form of software.[78] The practical effect of this decision was to free patent prosecutors from claiming a process that included some software implemented step. The patent prosecutors could claim software by wording the claim as the software implemented in a general purpose computer.

---

70. *Id.* at 187.

71. *Id.* at 191.

72. *Diehr*, 450 U.S. at 175.

73. Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Cal. L. Rev. 1, 9 (2001); *see also* FFII, *17,500 Software Patents to Issue in 1998*, http://swpat.ffii.org/penmi/bmwi-20000518/aharonian/stat-1998.txt (accessed Mar. 19, 2005).

74. *In re Pardo*, 684 F.2d 912, 915 (C.C.P.A. 1982); *see also Application of Freeman*, 573 F.2d 1237, 1246 (C.C.P.A. 1978); *Application of Walter*, 618 F.2d 758, 767 (C.C.P.A. 1980); In re *Abele*, 684 F.2d 902, 907 (C.C.P.A. 1982). First, the claim is analyzed to determine whether a mathematical algorithm is directly or indirectly recited. Next, if a mathematical algorithm is found, the claim as a whole is further analyzed to determine whether the algorithm is "applied in any manner to physical elements or process steps," and, if it is, it "passes muster under § 101." *In re Pardo*, 684 F.2d at 915 (quoting *Walter*, 618 F.2d 758, 767 (C.C.P.A. 1980)).

75. *Diehr*, 450 U.S. at 187. (inquiring whether there was any otherwise patentable subject matter contained within the claim) *Id.*

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

77. *Diehr*, 450 U.S. at 187. *See* In re *Alappat*, 33 F.3d at 1526.

78. *In re Alappat*, 33 F.3d at 1545.

Even after the *In re Alappat* decision, a patent applicant could not successfully directly claim software with no other otherwise patentable subject matter.[79] This ability became reality only four years later when the Federal Circuit, in a culmination of nearly twenty years of court precedent, ruled positively on the patentability of software in *State Street Bank & Trust Co. v. Signature Financial Group*.[80] The subject invention in *State Street* was a software application that managed a conglomerate mutual fund.[81] The invention did not contain the classical "otherwise patentable subject matter" that was key to a valid patent nor did it contain some tangible result.[82] This was simply a software application that produced share prices for a conglomerate mutual fund.[83]

In upholding the patentability of the software management system, the court reasoned that the computation of numbers was a tangible result and that even physical structure was unnecessary.[84] The court encouraged less focus on the type of patentable subject matter an invention was and more focus on whether the invention meets statutory muster by being novel, non-obvious, and useful.[85] The court held the *Freeman-Walter-Abele* test to be of little applicability in determining the existence of patentable subject matter.[86] Lastly, the court took the opportunity to abrogate the "ill-conceived" mythical business method exception to patentable subject matter.[87] Subsequent to *State Street*, pure software inventions are now patentable as long as they produce some new and useful result.[88] The software patent applicant would no longer be subjected to some tangible or physicality requirement to reach patentable subject matter.

However controversial the *State Street* decision was regarded, the courts were not dissuaded from their new interpretation of patentable subject matter. The Federal Circuit would later affirm the holding in *State Street* with *AT&T Corp. v. Excel Communications, Inc.*[89] The court

79. *Id.* at 1526.

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

81. *Id.* at 1371. The subject invention was a software-based system that maintained and managed a system for the sale and transfer of shares in a conglomerate mutual fund. This conglomerate mutual fund was composed of many smaller mutual funds that were configured in a spoke and wheel structure. The system kept track of the specific funds and to which hub they belonged allowing for the smooth flow of money in and out of the hub. *Id.*

82. *Id.*

83. *Id.*

84. *Id.* at 1373.

85. *State Street*, 149 F.3d at 1375.

86. *Id.* at 1374.

87. *Id.* at 1375-76.

88. *Id.*

89. *State Street*, 149 F.3d at 1368; *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352 (Fed. Cir. 1999).

rejected the premise that patentable software inventions must have a physical result and upheld the test of patentability promulgated in *State Street*.[90] The end result of twenty years of *Diehr* progeny was the unquestioned patentability of software in the U.S. and the definite end to the business method exception.[91]

IV. THE LAW OF THE EUROPEAN UNION

A. CONVENTION ON THE GRANT OF EUROPEAN PATENTS

Similar to the U.S., the EU has a centralized system for the granting and management of patent applications as provided by the Convention on the Grant of European Patents ("EPC").[92] The EPO was established under the EPC.[93] In most respects, the EPO is the EU equivalent of the PTO since all EU member states have adopted the EPC.[94] The EU-based inventor files only one application that if granted will garner a bundle of rights equivalent to individually issued patents in all EPC contracting nations.[95] One significant difference between the PTO and the EPO is that although the patent application process is unified under the

90. *AT&T*, 172 F.3d at 1360-61; *State Street*, 149 F.3d at 1368.

91. *State Street*, 149 F.3d at 1375-76.

92. *See* EPC, *supra* n. 3 (realizing the benefits of a centralized patent system and an unified procedure for obtaining patent protection in Europe, the signatories executed the EPC to fulfill these benefits and put themselves on par with the U.S.).

93. EPC, *supra* n. 3, art. 4, http://www.european-patent-office.org/legal/epc/e/ar4.html #A4 (accessed Mar. 19, 2004). The European Patent Organisation, as established by the EPC, is constituted by two main bodies: the executive body and the legislative body. The EPO is the executive body whereas the Administrative Council is the legislative body. EPO, *About the EPO*, http://www.european-patent-office.org/epo_general.htm#organ (accessed Mar. 19, 2005). The EPO is responsible for the issuance of patents for the contracting states; the activities of the EPO are regulated by the Council that is composed of delegates from the contracting states.

94. EU, *Proposal for a Directive on the Patentability of Computer-Implemented Inventions - Frequently Asked Questions*, http://europa.eu.int/comm/internal_market/en/indprop/comp/02-32.htm (accessed Mar. 19, 2005). "The EPC currently includes among its membership all [EU member states] plus five other countries (Switzerland, Turkey, Cyprus, Monaco, and Liechtenstein)." *Id.* The current members of the EU include the following nations: Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, The Netherlands, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, and United Kingdom. EU, *The Member States*, http://www.eurunion.org/states/home.htm (accessed Mar. 19, 2005). The current members of the EPC include the following nations: Austria, Belgium, Bulgaria, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia (expected to become member in due course), Liechtenstein, Lithuania, Luxembourg, Monaco, the Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey, and the United Kingdom. EPO, *Information from the European Patent Office*, http://www.european-patent-office.org/news/info/2004_10_11_e.htm (accessed Mar. 19, 2005).

95. EPC, *supra* n. 3, art. 2(2); EU, *supra* n. 94.

EPC, the enforcement and litigation of infringement is still within the province of the individual contracting states.[96] Consequently, although an EPO issued patent is valid throughout the EU, the patent's actual enforcement rights vary from nation to nation.

B. CURRENT LAW

The EPC's substantive law on patentable subject matter is contained within Chapter I of Part II of the EPC.[97] Article 52 lays out the statutory requirements for patentability in stating: "European patents shall be granted for any inventions which are susceptible of industrial application, which are new and which involve an inventive step."[98] The EPC spares the categorizations of U.S. patent law,[99] requiring only that the invention have industrial applicability, novelty, and inventive step. These statutory requirements mirror the U.S. statutory requirements of utility, novelty, and non-obviousness, respectively.[100] Rather than relying on court interpretation to render the exceptions to patentability, the EPC expressly forbids the patentability of "(a) discoveries, scientific theories and mathematical methods; (b) aesthetic creations; (c) schemes, rules and methods for performing mental acts, playing games or doing business, and *programs for computers*; (d) presentations of information."[101]

Upon cursory examination of Article 52, the casual reader could conclude the issue of patentability of software negatively; the more thorough reader would keenly note that the wide swath cut into patentability by paragraph two is steadied by paragraph three, which states: "(3) [t]he provisions of paragraph two shall exclude patentability of the subject-matter or activities referred to in that provision only to the extent to which an European patent application or European patent relates to such subject-matter or activities *as such*."[102] The last two words in paragraph three of Article 52 form the cornerstone for patentability of com-

96. EPC, *supra* n. 3, art. 2(2).

97. *Id.* arts. 52-57, http://www.european-patent-office.org/legal/epc/e/acii_i.html#AII_I (accessed Mar. 19, 2005).

98. *Id.* art. 52(1), http://www.european-patent-office.org/legal/epc/e/ar52.html#A52 (accessed Mar. 19, 2005).

99. 35 U.S.C. § 101 (2005); EPC, *supra* n. 3, art. 52. The *State Street* decision encouraged less emphasis on the classification of the subject invention and more emphasis on the other statutory requirements of patentability. *State St. Bank & Trust Co. v. Signature Fin. Group*, 149 F.3d 1368, 1375 (Fed. Cir. 1998). This de-emphasis on classifying patentable subject matter was a step toward the EPC stance on patentability.

100. 35 U.S.C. §§ 101-03 (2005).

101. EPC, *supra* n. 3, art. 52(2) (emphasis added).

102. *Id.* (emphasis added).

puter-implemented inventions[103] by the EPO Board and provide the legal wiggle room for the EPO to fit approximately twenty thousand software patents.[104]

The EPC equivalent of the U.S. Federal Circuit is the Board.[105] In a series of decisions by the Board, the patentability of software related inventions was established. The Board first addressed computer-implemented inventions in *In re Vicom Systems Inc.*; the Board upheld the patentability of a rejected application and held that "even if the idea underlying an invention may be considered to reside in a mathematical method [or any unpatentable subject matter under EPC Article 52(2)] a claim directed to a technical process in which the method is used does not seek protection for the mathematical method as such."[106] The subject invention in the *Vicom* decision was a method and apparatus for digitally processing images.[107] The Examining Division of the EPO ("Division") had rejected the application on three grounds: that the invention was an unpatentable mathematical formula under Article 52(2), that the invention was an unpatentable computer program "as such" ("computer program as such") under Article 52, and that the application failed to disclose the required technical features.[108]

The applicant appealed this decision to the Board on all three rejections. The Board held that the subject invention did contain proper patentable subject matter by reasoning that although the invention did not

103. The term software patents although popular is not the term of art used within the EU to reference software related inventions. The more common term is computer-implemented inventions. *See*, Software Patent Directive, *supra* n. 10.

104. FFII, *supra* n. 2. Some anti-software patent activist claim this interpretation by the EPO so dubious as to denigrate the democratic nature of the EU. *See* SWM Software-Marketing GmbH, *Current Patent Law*, http://www.nosoftwarepatents.com/en/m/politics/current.html (accessed Mar. 19, 2005). One especially virulent accusation stated "[the EPO] and their political friends, and large corporations with their special interests, want to elevate a dubious interpretation of the [EPC] to an EU directive." *Id.*

105. The Board is established by the EPC and responsible for "responsible for the examination of appeals from the decisions of the Receiving Section, Examining Divisions, Opposition Divisions and of the Legal Division." EPC, *supra* n. 3, art. 21, http://www.european-patent-office.org/legal/epc/e/ar21.html#A21 (accessed Mar. 19, 2005).

106. *In re Vicom Sys., Inc.*, 1987 E.P.O.R. 74 (Technical Bd. App. 1986), http://legal.european-patent-office.org/dg3/biblio/t840208ep1.htm (accessed Nov. 16, 2005).

107. *Id.* at 77. The subject invention has been issued a U.S. patent by the PTO. U.S. Patent No. 4,330,833 (issued May 18, 1982), http://patft.uspto.gov/netacgi/nph-Parser?u=netahtml/srchnum.htm&Sect1=PTO1&Sect2=HITOFF&p=1&r=1&l=50&f=G&d=PALL&s1=4330833.WKU.&OS=PN/4330833&RS=PN/4330833 (accessed Nov. 16, 2005). The issued patent succinctly described the invention as "[a]n improved method and apparatus for digital image processing is disclosed which permits greater efficiency in implementation of digital filtering techniques." *Id.*

108. EPC, *supra* n. 3, art. 52(3); *Vicom*, 1987 E.P.O.R. at 76. *See also* EPC, *supra* n. 3, arts. 52(2)(a), 29(1), http://www.european-patent-office.org/legal/epc/e/r29.html#R29 (accessed Mar. 19, 2005).

produce tangible or concrete results, the invention did have industrial applicability or utility.[109] This industrial and technological use of processing images was enough to remove the subject invention from the unpatentable coverage of computer program as such. The Board then addressed the Division's argument that the application related to unpatentable abstract mathematical formula. The Board rejected this premise finding the processing of digital images to be a tangible activity and espousing the rule that although an invention's underlying idea was unpatentable subject matter, a patent may be issued on the invention as far as it does not encompass the unpatentable idea.[110] The Board disposed of the Division argument that the invention constituted a computer program as such under similar reasoning.[111] After *Vicom*, a computer-implemented invention was patentable if its use was directed to some technical process; the patentable invention needed technical character.

The Board espoused a *Diehr*-like principle when it emphasized that an invention must be assessed as a whole in *Siemens A.G. v. Koch*.[112] The disposition of this case was an appeal of a rejected opposition filed by Siemens A.G.[113] The Board held that the EPC does not prevent patenting subject matter "consisting of a mix of technical and non-technical elements."[114] The Board stated that the invention's non-technical and technical elements should not be considered individually since any amount of technical element removed it from automatic proscription by Article 52(2)(c) and (3) of the EPC.[115]

Although the Board has yet to fully endorse computer-implemented inventions, in *In re Sohei*, the Board handed down its most favorable

---

109. *Vicom*, 1987 E.P.O.R. at 78.

110. *Id.* at 76, 78.

111. *Id.* at 76, 79.

112. *Diehr*, 450 U.S. at 175; *Siemens AG v. Koch & Sterzel*, 1988 E.P.O.R. 72 (Technical Bd. App. 1987), http://legal.european-patent-office.org/dg3/biblio/t860026ep1.htm (accessed Nov. 16, 2005).

113. Articles 99-105 of the EPC establish the grounds and procedure for an opposition proceeding. This allows any person to file with the EPO a request to invalidate the issued EPO patent. *See* EPC, *supra* n. 3, art. 99. Valid grounds for an opposition are laid out in article 100 and include: unpatentable subject matter, defective technical disclosure, or excessively broad claims that go beyond the actual invention. *Id.* art. 100, http://www.european-patent-office.org/legal/epc/e/ar100.html#A100 (accessed Mar. 19, 2005). In an opposition hearing, the owner of the issued patent and any other interested third parties can oppose the party that requested the opposition. *Id.* art. 99(4), http://www.european-patent-office.org/legal/epc/e/ar99.html#A99 (accessed Mar. 19, 2005). The PTO procedure that most resembles the opposition proceeding is reexamination. The PTO reexamination procedure does have significant differences such as being entirely post issuance whereas opposition can be requested during the prosecution process. *See* 35 U.S.C. § 302 (2005).

114. *Siemens*, 1988 E.P.O.R. at 76.

115. *Id.*

opinion for the patentability of computer-related inventions.[116] The appeal arose from the Division's rejection of an application covering a computer-implemented inventory and financial management system.[117] In *re Sohei* held that patentability cannot be destroyed by the addition of an element that would be excluded as unpatentable under Article 52 subparts (2)(c) and (3).[118] The Board reasoned that an invention that implements functional elements through the use of software was not unpatentable under Article 52 of the EPC if the invention solves a problem that is necessarily technical in nature.[119] The Board argued that the technical nature of the problem solved suggested an implicit technical nature in an invention that was not expressly technical.[120]

Subsequently, the Board held, in *In re Pension Benefit*, the contrapositive[121] of the *In re Sohei* ruling when it stated that the addition of a technical element to a non-technical invention did not remove the invention from unpatentable subject matter under Article 52. The Board also upheld business method exception to patentability.[122] The Board further endorsed computer-implemented inventions when it redefined the technical nature requirement for these types of inventions.[123] The Board held that although a computer-implemented invention was not per se technical in nature, the technical nature requirement could be satisfied through certain further "effects" of a technical nature.[124] These requisite effects could be satisfied within the internal operation of the program. The Board went on to state:

A patent may be granted not only in the case of an invention where a piece of software manages, by means of a computer, an industrial process or the working of a piece of machinery, but in every case where a program for a computer is the only means, or one of the necessary means, of obtaining a technical effect [nature] within the meaning specified above, where, for instance, a technical effect [nature] of that kind is achieved by the internal functioning of a computer itself under the influence of the

116. *In re Sohei*, 1996 E.P.O.R. 253 (Tech. Bd. App. 1994), http://legal.european-patent-office.org/dg3/biblio/t920769ep1.htm (accessed Mar. 19, 2005).

117. *Id.*

118. *Id.* at 259.

119. *Id.* at 258.

120. *Id.*

121. A contrapositive is a proposition derived by negating and permuting the terms of another, equivalent proposition; for example, All not-Y is not-X is the contrapositive of All X is Y. *See* Wikipedia, *Contrapositive*, http://en.wikipedia.org/wiki/Contrapositive (accessed Nov. 16, 2005).

122. *In re Pension Benefit Sys. P'ship*, 2002 E.P.O.R. 522, 528 (Tech. Bd. App. 2000), http://legal.european-patent-office.org/dg3/biblio/t950931eu1.htm (accessed Mar. 19, 2005).

123. *In re IBM Corp.*, 1999 E.P.O.R. 301 (Tech. Bd. App. 1999), http://legal.european-patent-office.org/dg3/biblio/t970935eu1.htm (accessed Mar. 19, 2005).

124. *Id.* at 310.

said program.[125]

The Board went on to hold that all computer programs that have the requisite "technical effect" were patentable subject matter within the meaning of Article 52(1) if otherwise patentable under the EPC.[126] The end result of this string of Board decisions was that computer-implemented inventions were patentable under the EPC if the invention meets the technical nature requirement.

C. Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions

Since the EPO contracting states were free to develop their own national laws for the enforcement and litigation of EPO issued patents, incongruence amongst the EPC contracting states has developed in regards to enforcement of computer-implemented invention patents.[127] Consequently, a computer-implemented invention patent issued by the EPO has varied effectiveness throughout Europe. Although the EPC is a separate entity from the EU, all EU member states are contracting states of the EPC. Thus, a problem within the EPC consortium is a problem for the entire EU.[128] This incongruence within the EU injected ambiguity and uncertainty for patent holders in Europe and drew the attention of the pro-software patent software industry and anti-software patent activists alike.[129] Regardless of their viewpoint, the common demand from both sides was for clarity and certainty.[130]

As early as 2002, the EU realized the issue of computer-implemented invention patents demanded continent wide discussion and ordered a study be conducted on the "The Economic Impact of Patentability of Computer Programs."[131] After a consultation period, the European Commission drafted and presented the Software Patent Directive.[132] The Software Patent Directive's intent was to harmonize the national patent laws of EU member states and codify established EPO practice in regards to computer-implemented invention patents.[133] The Software Patent Directive defined a computer-implemented invention as "any in-

125. *Id.*
126. *Id.*
127. *See* EU, *supra* n. 94.
128. *Id.*
129. *See* Software Patent Directive, *supra* n. 10.
130. *Id.*
131. EU, *Study "The Economic Impact of Patentability of Computer Programs"*, http://europa.eu.int/comm/internal_market/en/indprop/comp/studyintro.htm (accessed Mar. 19, 2005).
132. Software Patent Directive, *supra* n. 10.
133. *Id.* at 3, Explanatory Memorandum.

vention the performance of which involves the use of a computer, computer network or other programmable apparatus and having one or more prima facie novel features which are realized wholly or partly by means of a computer program or computer programs"[134] and required member states to implement patent laws to protect them.[135] The Software Patent Directive adopted the current position of the EPO Board by conditioning patentability of computer-implemented inventions on a "technical contribution".[136]

With a draft of the prospective legislation ready, the Software Patent Directive was injected into the EU co-decision legislative process.[137] Three years after the European Commission's proposed Software Patent Directive and several versions later,[138] the Council approved of an amended version of the Software Patent Directive and submitted it to the European Parliament for final approval.[139] Once approved by the European Parliament, the Software Patent Directive would have been on its way to becoming legislation throughout the EU.

Unfortunately, the political pressure from several groups was too much for the Software Patent Directive to bear. In July of 2005, the European Parliament, citing difficulties in finding a version that pleased all parties, voted to reject the Software Patent Directive.[140] After the rejection of the Software Patent Directive, the European Commission indi-

134. *Id.* art. 2(a).

135. *Id.* art. 9.

136. *Id.* art. 4.

137. Alex Hudson, *The Software Patent Directive: Progress in Europe*, http://www.softwarepatents.co.uk/current/ (accessed Mar. 19, 2005). The chart, *see id.*, has been updated to incorporate recent developments in the co-decision process. *See* BBC, *infra* n. 139. The EU has three main institutions that drive the process of legislation: the European Commission ("Commission"), the European Parliament ("Parliament"), and the Council of the European Union ("Council"). The Commission's primary role is to propose drafts of legislation whereas the Parliament and Council are responsible for the passage of the draft into legislation. There are three processes, distinguished by how the Council and Parliament interact, for passage of legislation within the EU: co-decision, consultation, and assent. The Software Patent Directive lies within the subject matter jurisdiction of the co-decision process. In the co-decision process, the Commission sends the draft legislation to both the Council and Parliament for two readings and amendments. EU, *Decision-making in the European Union*, http://www.europa.eu.int/institutions/decision-making/index_en.htm (accessed Mar. 19, 2005).

138. For a brief yet exhaustive discussion of the events between the proposed Software Patent Directive submitted by the Commission and the passed amended version by the Council, see Wikipedia, *Directive on the Patentability of Computer-Implemented Inventions*, http://en.wikipedia.org/wiki/Proposed_EU_Directive_on_the_patentability_of_computer-implemented_inventions (accessed Mar. 20, 2005).

139. BBC, *EU Ministers Endorse Patent Law*, http://news.bbc.co.uk/1/hi/technology/4325215.stm (Mar. 7, 2005).

140. BBC, *Software Patent Bill Thrown Out*, http://news.bbc.co.uk/2/hi/technology/4655955.stm (July 6, 2005).

cated that it would not draft up a new version; the Software Patent Directive was officially dead.[141]

V. COMPARATIVE ANALYSIS BETWEEN THE UNITED STATES AND THE EUROPEAN UNION

Whether called software patents or computer-implemented invention patents, the right to these patents is firmly established on both sides the Atlantic; the number of software patents issued by the EPO and the PTO exhibit this reality.[142] The U.S. has fully adopted the patentability of software through judicial evolution of patentable subject matter.[143] In the U.S., the software patent applicant need only submit a software invention that produces a tangible result. The U.S. courts provided further approval of software patents when "a useful, concrete, and tangible result" was defined as possibly being intangible and without physical presence.[144] The U.S. abrogated the business method exception to patentability;[145] thus, the infamous software-implemented business method patent became a reality.[146] Now, software patents in the U.S. can issue for methods that were wholly non-technical ways of doing business.

In contrast to U.S. courts, the EPO Board has yet to fully endorse software patents.[147] The Board has maintained that to remain patentable, a computer-implemented invention must possess some technical nature or effect. A lack of technical contribution would cause the invention to be encompassed by the EPC Article 52 proscription of computer program as such patents. The EPO has also declined to abrogate the business method exception as the U.S. did in *State Street*.[148] Although the Board has provided an unified position on the issuance of computer-implemented invention patents throughout Europe, the effectiveness of enforcement of EPO issued patents is varied at best.[149] Conversely, the

141. *Id.*

142. McMillan, *supra* n. 1; FFII, *supra* n. 2.

143. *See* discussion *supra*, Part II.

144. *State St. Bank & Trust Co. v. Signature Fin. Group*, 149 F.3d 1368, 1373 (Fed. Cir. 1998).

145. *Id.* at 1375-77.

146. *See* U.S. Patent No. 6,389,458 (issued May 14, 2002) (pop-up windows patent); U.S. Patent No. 6,289,319 (issued Sept. 11, 2001) (paying with a credit card online patent); U.S. Patent No. 6,026,368 (issued Feb. 15, 2000) (targeted banner advertising patent); U.S. Patent No. 5,960,411 (issued Sept. 28, 1999) (Amazon one-click online shopping patent); U.S. Patent No. 5,715,314 (issued Feb. 3, 1998) (online shopping cart patent); U.S. Patent No. 4,873,662 (issued Oct. 10, 1989) (hyperlink patent).

147. *See* discussion *supra*, Part III.

148. *State Street*, 149 F.3d at 1373; In re *Pension Benefit Sys. P'ship*, 2002 E.P.O.R. 522, 528 (Tech. Bd. App. 2000).

149. EU, *supra* n. 94.

enforcement of software patents is firmly established throughout the U.S. Although software firms can seek and receive issuance of similar software patents in the U.S. and the EU, there is a significant chance that the EU patent will not be enforceable throughout Europe.

VI. CRITICAL ANALYSIS ON SOFTWARE PATENTABILITY

A. ENDING THE TYRANNY OF BAD SOFTWARE PATENTS

Since the U.S. has taken the most liberal stance on software patentability, providing for uniform enforcement of software patents, opponents of software patents commonly cite notorious patents issued by the PTO, foreshadowing the effects of adopting a similar position in the EU.[150] Adopting such a position is not only erroneous but fails to grasp the bigger issue. These "bad [software] patents" do not arise intrinsically from the patentability of software.[151] The majority of these "bad software patents" do not meet the basic statutory requirements for patentability prescribed by the U.S. Congress.[152] These improperly issued software patents are the result of defective PTO examining procedure.[153]

One particularly notorious example is the Amazon one-click software patent.[154] This is an excellent example of a patent that should have never been issued. This patented invention was arguably anticipated by the prior art existing at the time. Thus, the invention lacked the necessary novelty mandated by U.S. statute.[155] Even if the invention theoretically met the novelty requirement, the invention's novel contributions were obvious to the person of ordinary skill in the pertinent

150. *See,* Electronic Frontier Foundation ("EFF"), *Patent Busting's Most Wanted,* http://www.eff.org/patent/wanted/ (accessed Mar. 19, 2005); FFII, *Software Patents in Action,* http://swpat.ffii.org/patents/effects/index.en.html#kazoj (accessed on Mar. 21, 2005).

151. *Patent Nonsense,* The Economist, Apr. 6, 2000, http://www.economist.com/ displayStory.cfm?Story_ID=300582 (accessed Mar. 19, 2005).

152. *See* 35 U.S.C. §§ 101-04 (2004).

153. Economist, *supra* n. 151.

154. U.S. Patent No. 5,960,411 (issued Sept. 28, 1999), http://patft.uspto.gov/netacgi/nphParser?TERM1=5960411&u=/netahtml/srchnum.htm&Sect1=PTO1&Sect2=HITOFF&p=1&r=0&l=50&f=S&d=PALL (accessed Nov. 16, 2005). *See also* Free Software Foundation, *Boycott Amazon!,* http://www.gnu.org/philosophy/amazon.html (accessed Mar. 21, 2005); Evan Ratliff, *Patent Upending,* Wired Magazine, June 2000, http://www.wired.com/wired/archive/8.06/patents.html (accessed Nov. 16, 2005).

155. Novelty is one of the statutory requirements for patentability in the U.S. and is equivalent to lack of anticipation. An invention is considered anticipated by the prior art when the contributions of the invention are taught by the prior art. *See* Black's Law Dictionary, *supra* n. 46. The invention is anticipated when every element of the accompanying claims are either expressly or inherently, to a person skilled in the pertinent art, contained within a single prior art reference. Only one prior art reference can anticipate the invention. An invention anticipated by the prior art is not patentable in the US. *See* 35 U.S.C. § 102(a) (2004).

art.[156]

One common cause cited for the rash of dubious software patents issued by the PTO is lack of prior art in the PTO.[157] Since the U.S. just recently acknowledged the legitimacy of software patents and business method patents, the PTO lacks a library of relevant issued patents that have expired or been abandoned to compare the patent application's disclosed invention against. The logical solution to this lack of PTO housed prior art is to make it incumbent upon the examiner to seek outside sources of prior art. This simple act of ascertaining the current state of prior art would eliminate the issuance of most bad software patents. Current PTO examining procedures assume that expired patents represent the state of the art. This is obviously not the case with software and business method patents. Unfortunately, the PTO is notoriously overworked and the average time spent per application is invariably too short to allow the examiner to undertake an exhaustive search of the extra-PTO prior art.[158] During the 1990s, the number of patent applications nearly doubled and the examiner work force did not keep up.[159] As a result, the number of examiners per thousand applications had decreased by twenty percent by the end of the decade.[160] The PTO culture that rewards quantity over quality exacerbates the problem by encouraging the examiner to spend even less time on the patent application.[161]

One ancillary effect of the increased examiner workload is increased average pendency.[162] Pendency is the time between filing and issuance or abandonment of utility, plant, and reissue applications. The PTO's average pendency for software patent applications is over three years.[163] Since only prior art before the filing is considered in the issuance of a patent, long pendency can create the problematic scenario of the submarine patent.[164] The U.S. Congress attempted to address the problem of

156. Non-obviousness is one of the statutory requirements for patentability in the U.S. An invention deemed sufficiently novel may still fail to meet statutory requirements for patentability if the differences between the invention and the prior art are obvious to a person of ordinary skill in the pertinent art. In determining the non-obviousness of an invention, multiple references in the prior art may be considered. *Id.* § 103.

157. *See*, The Economist, *supra* n. 151.

158. Natl Acad. of Sciences, *A Patent System for the 21st Century* 51-52 (Stephen A. Merill et al. eds., 2004), http://www.nap.edu/books/0309089107/html (accessed 19, 2005).

159. *Id.* at 51.

160. *Id.*

161. *See e.g.*, The Economist, *supra* n. 151.

162. PTO, 2003 Performance and Accountability Report 106 (2004), http://www.uspto.gov/web/offices/com/annual/2003/2003annualreport.pdf (accessed Mar. 19, 2005).

163. *Id.* at 109.

164. Submarine patents are issued patents that had an exceptionally long pendency periods and issue after the invention has become commonly used in the relevant industry. The industry is placed in a weak bargaining position after investing in the invention's technology; thus, the patentee subsequently issues licensing agreements with favorable terms

the submarine patent when it mandated publication of pending patents eighteen months after the filing of the patent application.[165] Although this legislative fix is of limited effectiveness, the patent applicant can request the application not be published if there is no intent to secure foreign patents on the subject invention.[166] Given many software patents, especially software-implemented business method patents, are not patentable in major foreign markets, the software patent applicant can request confidentiality of the application with little to no loss.[167]

The solution to the PTO's ailments is clear; the PTO must change the culture within it to encourage the examiners to spend more time searching out the relevant prior art regardless of the location and reward the examiners for producing patents of high quality instead of the quantity of disposed applications. The software industry is a fast paced emerging field where most prior art is located in myriad of poorly indexed sources such as professional journals, academic journals, conference proceedings, whitepapers, marketing presentations, Web sites, Usenet postings, and so forth. The U.S. Congress must authorize increased funding for the PTO to hire more examiners to decrease the overall workload and reduce excessive pendency. More simply, the U.S. Congress should appropriate all PTO generated fees for the operation of the PTO and cease siphoning off PTO funds for other purposes.[168] These actions will reduce workload on the average examiner and enable them to properly search all of the prior art and prevent the issuance of bad software patents.

The PTO should also implement a policy requiring the publication of patent applications at the time of filing with no exceptions, ensuring the software industry is put on notice of patents in the prosecution pipeline. This policy will enable the software industry to prepare and collect possible prior art for future validity challenges. Currently, the PTO does not allow for third party challenges during the prosecution process, similar

---

to the firms invested in the invention. Black's Law Dictionary (Bryan A. Gardner Ed., 8th ed., West 2004).

165. 35 U.S.C. § 122(b)(1)(A) (2005).

166. *Id* at. § 122(b)(2).

167. Robert Hart Et al., *Intellectual Property Institute, The Economic Impact of Patentability of Computer Programs* 2 (2000), http://europa.eu.int/comm/internal_market/en/indprop/comp/study.pdf (last visited Mar. 31, 2005).

168. American Bar Association, *PTO Fee Diversion Costs Jobs (2003)*, http://www.abanet.org/intelprop/feediversion.pdf (accessed Mar. 31, 2005). The U.S. Congress did consider the cessation of PTO fee diversion in 2004. The U.S. Congress incorporated the proposed changes into the Consolidated Appropriations Act of 2004, *see* Consolidated Appropriations Act of 2004, Pub. L. No. 108-356, 118 Stat. 3. Unfortunately, the U.S. Congress passed the bill with the anti-diversion language removed. *See* American Bar Association, *2005 Legislative Priorities: US Patent and Trademark Office Funding*, http://www.abanet.org/poladv/priorities/patent.html#status (accessed Mar. 26, 2005).

to the opposition process within the EPO.[169] A less drastic change to current PTO procedure would be to simply allow third parties to submit relevant prior art for discretionary examiner consideration. A discretionary consideration of third party provided prior art compliments the policy of patent application publication.

There is a tenable argument that this problem will likely solve itself over time. Since the patentability of software has only been established in the U.S. for seven years and with patent terms lasting twenty years from filing, software patents will not begin falling into the public domain for several years.[170] Eventually,[171] the in-house prior art at the PTO will expand and current PTO procedures hypothetically will suffice. Nonetheless, the implementation of these preceding policies would certainly reduce the number of anticipated or obvious patents being issued and the number of possible submarine patents in the near future.

B. Software Patents: Impediments or Catalysts of Innovation

Opponents of software patents regularly argue that software patents will reduce competition and slant the competitive field of play in favor of the large corporations.[172] They contend that most software patents are issued to large software firms whom possess rather expansive and expensive patent portfolios.[173] Patent prosecution is an expensive proposition, with fees for securing a patent typically exceeding one thousand dollars, even for a small entity.[174] Consequently, these opponents argue that the small software firms cannot build up a patent portfolio and will be driven insolvent by paying royalties to larger firms and being exposed to the even more costly prospect of patent litigation.[175]

---

169. *See* EPC, *supra* n. 3, arts. 99-105.

170. 35 U.S.C § 154(a)(2) (2005); *State St. Bank & Trust Co. v. Signature Fin. Group*, 149 F.3d 1368 (Fed. Cir. 1998).

171. Given that the actual useful term of software patents is much shorter than twenty years due to the fast pace at which the software art changes, many software patents will expire due to failure to pay maintenance fees or abandonment by patent holder. Peter J. Toren, *Patent Problems? The Solution IsÃ‚A. . .*, http://library.findlaw.com/2000/Sep/1/130447.html (accessed Nov. 16, 2005).

172. FFII, *Protect Innovation*, http://protectinnovation.ffii.org.uk/ (accessed Mar. 22, 2005).

173. FFII, *supra* n. 2. The most active applicants for software patents in Europe include the likes of Matsushita Electric Industries Co., Microsoft Corp., and Sony Corp.

174. PTO, *PTO Fiscal Year 2005 Fee Schedule (2004)*, http://www.uspto.gov/ web/offices/ac/qs/ope/fee2004dec08.htm (accessed Nov. 16, 2005). The PTO has a second tier decreased fee scale for small entities. *Id.*

175. The average cost of patent litigation is two million dollars. William Robinson, *IP Litigation Strategies: Patents: Markman Hearings*, http://library.findlaw.com/2003/Sep/30/133071.html (accessed Nov. 16, 2005).

The general rule is that patents engender competition and innovation, and software patents are not an exception. Although software patents may invariably lead to increased costs for software firms, there are certain measures that can mitigate this problem. Patent prosecution costs could be reduced for the smaller software firms; the U.S. Congress has already acted in this regard by passing the Consolidated Appropriations Act of 2004 that revises the PTO fee schedule by reducing small entity patent filing fees.[176] The costs of patent litigation are much more difficult to remedy. Although the burden of prospective patent litigation is substantial for small software firms, this burden is borne by all industries. Although arguably a treatment of the symptom and not the illness, one recent solution is that software firms purchase some form of patent insurance to protect against prohibitively expensive patent litigation.[177] Another solution that better addresses the problems of excessive costs and excessive frequency of patent litigation is to create a specialized U.S. federal district court to handle patent issues.[178] Patent litigation makes up only a small portion of the typical federal district court docket, a specialized court would better handle the technical issues that arise frequently in patent litigation and would provide predictability and consistency for patent litigants. Consequently, the abuse and frequency of patent litigation would decrease and make software patent litigation a less costly proposition for the software industry.

The most persuasive argument in favor of software patents is the success of the U.S. software industry under the software patent regime. Although the U.S. is the most liberal major economic nation in regards to software patents, the U.S. software industry accounts for nearly one half of the world software industry revenue.[179] The growth of the U.S. software industry is consistently outpacing the world and is projected to maintain this lead in the future.[180] Statistics indicate that software patents do not cripple the growth and success of the industry. More telling and contrary to anti-software patent rhetoric, the statistics indicate that software patents do not enable large corporate software firms to drive

176. Consolidated Appropriations Act of 2004, Pub. L. No. 108-356, 118 Stat. 3.

177. Steven J. Vaughan-Nichols, *Linux Companies Address IP Concerns*, Eweek, http://www.eweek.com/article2/0,1759,1631335,00.asp (Aug. 4, 2004).

178. Rather than the traditional geographical jurisdiction of the U.S. federal courts, this patent court would have exclusive subject matter jurisdiction over patent litigation actions. This form of subject matter jurisdiction is similar to the Court of Appeals for the Federal Circuit. Wikipedia, *United States Court of Appeals for the Federal Circuit*, http://en.wikipedia.org/wiki/United_States_Court_of_Appeals_for_the_Federal_Circuit (accessed Mar. 24, 2005).

179. Software & Information Industry Association, *Software Industry Statistics for the Third Quarter 2004*, http://www.siia.net/software/pubs/statpage_Q304.pdf (accessed Nov. 16, 2005).

180. *Id.*

smaller firms into nonexistence. Ninety-two percent of the firms in the U.S. software industry employs less than forty-nine employees.[181] Evidently, the impending doom for small software firms has yet to arrive in the seven years of unquestioned software patentability.

C. COPYRIGHT IS NOT THE PROPER PROTECTION SCHEME FOR SOFTWARE

Opponents of software patents regularly argue that patent protection for software is redundant and the wrong form of intellectual property protection.[182] Software intellectual property practice has evolved over the last half century. Copyright has not always been the preferred way to protect software.[183] As the software industry has evolved, the choice of intellectual property has evolved along with the industry.[184] Software firms have battled in court over the years to expand the protection they receive from copyright.[185] This has lead to many cases where wily competitors of innovative software firms have stolen ideas and escaped liability for copyright infringement.[186]

The run of copyright regime for software has been exhausted. Software firms have approached copyright litigation with patent-like protection arguments based upon the incompatible copyright law.[187] This has lead many courts to expressly lament the failures of copyright protection and ask for legislative intervention to avert intrinsically unfair results in copyright infringement litigations.[188] Courts have developed complex pyramid abstraction schemes while "attempt[ing] to fit the

---

181. Software & Information Industry Association, Software Industry Profile June 2004, http://www.siia.net/software/pubs/profile_0604.pdf (2004).

182. SWM Software-Marketing GmbH, *The Basics*, http://www.nosoftwarepatents.com/en/m/basics/index.html (accessed Mar. 19, 2005).

183. *See* discussion, *supra* Part II.

184. *Id.*

185. *See Softel, Inc. v. Dragon Med. and Scientific Communications, Inc.*, 118 F.3d 955 (2d Cir. 1997); *Lotus Dev. Corp. v. Borland Intl., Inc.*, 49 F.3d 807 (1st Cir. 1995) aff'd 516 U.S. 233 (1996); *Computer Assoc. Intl., Inc. v. Altai, Inc.*, 982 F.2d 693 (2d. Cir. 1992); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983).

186. *See Altai*, 982 F.2d at 693.

187. *See id.* at 701-02.

188. *See id.* at 712.

> To be frank, the exact contours of copyright protection for non-literal program structure are not completely clear. . . .Indeed, it may well be that the Copyright Act serves as a relatively weak barrier against public access to the theoretical interstices behind a program's source and object codes. This results from the hybrid nature of a computer program, which, while it is literary expression, is also a highly functional, utilitarian component in the larger process of computing. Generally, we think that copyright registration—with its indiscriminating availability—is not ideally suited to deal with the highly dynamic technology of computer science. Thus far, many of the decisions in this area reflect the courts' attempt to fit the proverbial square peg in a round hole.

Id.

proverbial square peg in a round hole" to properly filter idea from expression.[189] These problems are merely the fruition of the improper fit that is copyright protection for software. The content and arrangement of every line of code within software is an intrinsically useful and dependent on that use. Commentators have suggested that this intrinsic usefulness removes software completely from the realm of copyright.[190]

Although patent protection for software may represent the sought after "round peg", the size of the peg must be tailored properly to fit.[191] Metaphors aside, software patent enforcement rights should be fitted to the exigencies of the software industry. In congruence with recent judicial precedent from the Supreme Court of the U.S., the doctrine of equivalents should be weakened in application to software patents.[192] By narrowing the protective reach of software patents, the right to practice programming will be protected, and the primary concern of many software patent opponents will be addressed.

D. The Alleged Threat to the Open Source Community

The worldwide open source[193] community arguably represents the

---

189. *Altai*, 982 F.2d at 712.

190. *See* Dennis S. Karjala, *The Relative Roles of Patent and Copyright in the Protection of Computer Programs*, 17 J. Marshall J. Computer & Info. L. 41, 53-55 (1998).

191. *Altai*, 982 F.2d at 712.

192. *Graver Tank & Mfg. Co. v. Linde Air Products Co.*, 339 U.S. 605, 608 (1950) (explaining that the doctrine of equivalents is a judicially created basis for finding non-literal infringement of patents. The patent holder claims non-literal infringement through the doctrine of equivalents when the subsequent invention performs the substantially same function in a substantially similar way to the patented invention). In accordance with U.S. jurisprudence, the doctrine of equivalents can also be applied in reverse.

Thus, where a device is so far changed in principle from a patented article that it performs the same or a similar function in a substantially different way, but nevertheless falls within the literal words of the claim, the doctrine of equivalents may be used [in reverse] to restrict the claim and defeat the patentee's action for infringement. *SRI Int.l v. Matsushita Elec. Corp. of America*, 775 F.2d 1107, 1123 (Fed. Cir. 1985).

The Supreme Court of the U.S., in overruling the Federal Circuit, held that amendments made during the patent prosecution process surrender all equivalents between the prior rejected claim and the amended version of the claim unless: the equivalent was unforeseeable at the time of claim amendment or the equivalent was tangentially related to the amendment. *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 535 U.S. 722, 740 (2002). The *Festo* decision is representative of a current judicial trend in the U.S. to weaken the doctrine of equivalents. *See generally Warner-Jenkinson Co. v. Hilton Davis Chemical Co.*, 520 U.S. 17 (1997).

193. Cohen, *supra* n. 26, at 205. (discussing that the open source movement is predicated on the GNU General Public License ("GNU GPL") and the GNU GPL is a licensing system that authorizes people to use, modify, and redistribute the programs and to create and distribute new programs based on the initial program; the license requires that those new programs must be distributed subject to the same GNU GPL).

largest and most organized opponent to software patents.[194] The open source community's opposition is predicated on the fear that software patents may sound the death knell for their movement.[195] This fear is stoked by seemingly frivolous litigation against the most popular software application under the GNU GPL: Linux Operating System ("GNU GPL").[196] Open source advocates fear that if the legal standing of the GNU GPL software applications is called into question, businesses across the industrial spectrum will avoid these products forcing the GNU GPL into enterprise oblivion.

The current crusade against GNU GPL is powered not by allegations of software patent infringement but by copyright infringement.[197] Any form of intellectual property litigation presents an unique problem for the GNU GPL since there is no proprietor of the intellectual property. Without a prototypical owner and proprietor, there is no entity with a vested monetary interest in defending the legal standing of the GNU GPL. Software patents provide several benefits over traditional copyright protection in regards to the open source movement. Software patents expire much sooner than copyright and require proactive maintenance fees over their duration to keep them valid. Consequently, many software patents will be abandoned well before the twenty year term expires, whereas copyright terms last many times longer and require no affirmative action by the copyright owner to acquire or maintain. Moreover, the copyright holder must intentionally abandon the copyright in order to put it in the public domain.[198] Software patents also provide certainty in what is within the public domain since issued patents, expired patents, and abandoned patents are publicly available whereas, the copyright holder need not register nor publish copyrighted material to maintain protection. Additionally, copyright protection vests with no examination for originality or progress whereas software patents should only issue when statutory patentability requirements are met. Overall, properly issued software patent protection presents less of a threat to the open source movement than copyright protection.

VII. CONCLUSION

Software patents in the U.S. and the EU are established, although more extensively in the U.S. . The U.S. extended patentability to

194. EuroLinux Alliance for a Free Information Infrastructure, *About Eurolinux*, http://www.eurolinux.org/about/index.en.html (accessed Mar. 26, 2005); Open Source Technology Group, Inc., *About This Site*, http://slashdot.org/about.shtml (accessed Mar. 26, 2005).

195. *Id.*

196. *See* Spencer Ante, *A Linux Nemesis on the Rocks*, 3924 Business Week 80 (Mar. 14, 2005).

197. *Id.*

198. *See, A & M Records, Inc. v. Napster, Inc.*, 239 F.3d 1004, 1026 (9th Cir. 2001).

software through a series of judicial decisions culminating with an emphatic endorsement of its patentability in *State Street*. The EU has approached software patentability more cautiously through a series of decisions by the Board. The Board has maintained that software patents must contain some form of technical element to be valid patentable subject matter. Nonetheless, software patents issued by the EPO are of dubious value due to the varied enforcement of them by the law of EU member states. Consequently, the EU had proposed a Software Patent Directive to unify EU member states on their position on software patents. The last version of the Software Patent Directive would have codified the current EPO practice on software patents. If passed into law, the proposed EU Software Patent Directive would not have approached the liberal standard of the U.S. by maintaining the technical element requirement and maintaining the unpatentable nature of business methods.

Software patents are the most effective method of intellectual property protection for software available. The current rash of bad software patents in the U.S. is the result of poor PTO examining procedure and practice. This current difficulty can be addressed by: changing the PTO culture of quantity over quality, ensuring the quality of examining procedure so only novel, non-obvious, and useful patents issue, reducing the overall workload for examiners, requiring examiners to look at a broader prior art library, and requiring publication of all applications with no exceptions. This Comment is not alone in calling for these reforms.[199] There exists a consensus on most of the solutions. The only lacking element is political will among legislators.

Software patents engender growth and fair competition within the software industry as demonstrated by the statistics. Although the software industry has experienced tremendous growth over the last decade, certain measures should be made to maintain the diversity of firm size within the industry and maintain healthy competition. PTO fees for small entities should be reduced, and the growing fear of patent litigation will be addressed by the growing market for intellectual property insurance and the creation of a dedicated patent district court.

Although copyright has been the prominent form of intellectual property protection for software over the last twenty years, the time for a change has come. Copyright theory has been exhausted in its application to software, and patent protection has increasingly become the clear replacement for copyright to protect the intrinsically useful intellectual

199. *See* Microsoft Corp., *Q&A: Microsoft Calls for Reforms to the U.S. Patent System*, http://www.microsoft.com/presspass/features/2005/mar05/03-10PatentReform.asp (accessed Mar. 10, 2005); Chris Oakes, *Amazon Calls for Patent Fix*, Wired, http://www.wired.com/news/ technology/0,1282,34887,00.html (accessed Mar. 26, 2005).

property of software. Arguments from the open source community that software patents threaten the GNU GPL make little sense as the GNU GPL has more reason to fear copyrights rather than a properly issued software patents.

Although software patents have had their problems in the U.S., these problems arise not from software patentability but from other sources. The fast paced software industry is better served by patent protection rather than the copyright, as demonstrated by the unfettered growth of the U.S. software industry. The Software Patent Directive represented a well-thought intention to bring the EU in congruence with the U.S. in regard to software patentability with certain caveats. If the Software Patent Directive had been passed and implemented properly, the EU could have reaped the benefits of software patentability and avoided the implementation problems that currently plague the U.S. Unfortunately, the EU has chosen to keep their respective patent jurisprudence incongruent with that of the U.S. By rejecting the Software Patent Directive, the European Parliament handicaps their own software industry for years to come.