University of Massachusetts Law Review

Volume 3
Issue 1 *Trends and Issues in Intellectual Property*

Article 4

January 2008

Computer Programs Under the United States Intellectual Property System: Sui Generis Legislation is Needed

Joseph Francis Agnelli, III

COMPUTER PROGRAMS UNDER THE UNITED STATES INTELLECTUAL PROPERTY SYSTEM: SUI GENERIS LEGISLATION IS NEEDED

JOSEPH FRANCIS AGNELLI, III*

INTRODUCTION

Computers and computer software command an ever expanding and prominent role in the world, and are constantly evolving. Thus, various international governments are fraught with enacting appropriate legislation associated with protection of computer software. Should the United States Congress use copyright law, patent law, or a measure of both? This is the predicament Congress has been faced with regarding the rapid evolvement of computer software. This article will attempt to resolve that dilemma.

Section I of this article explores the different avenues of intellectual property protection presently available for computer software[1] here in the United States. Section II then discusses how the European Community has resolved the computer program crisis under European intellectual property law. Lastly, section III will illustrate why *sui generis*[2] legislation would be the paramount way for Congress to attack the intricacy that is created by computer programs under American intellectual property law.

---

*The author is an attorney and 2007 graduate of the Southern New England School of Law. All opinions and/or inaccuracies are of this writer, and do not represent the opinions or views of the Faculty or Students of the Southern New England School of Law, or the Southern New England School of Law Roundtable Symposium Law Journal.

[1] The terms "software" and "program" will be used interchangeably throughout this article.

[2] *Sui Generis* has been defined as a body of law which is "[O]f its own kind or class; unique or peculiar." BLACKS LAW DICTIONARY 1475 (8th Ed., 2004).

I. PROTECTION FOR COMPUTER PROGRAMS UNDER UNITED STATES INTELLECTUAL PROPERTY JURISPRUDENCE

This section will scrutinize the present state of intellectual property protection for computer programs in the United States. This section will begin with a discussion and analysis of copyright law, the simpler and more applicable, and will conclude with an analysis of patent law, the complex and controversial. The goal of this section is to help you, the reader, fully understand the evolution and problems that computer programs pose to Congress and, more importantly, to the courts. Hopefully, when you, the reader, are finished with this section, you will realize that action in the form of *sui generis* legislation needs to be taken to bring protection for computer programs into focus in American intellectual property jurisprudence.

A. United States Copyright Law as Applied to Computer Programs.

> "From its beginning, the law of copyright has developed in response to significant changes in technology."[3]

Throughout relevant history, United States Copyright law emerged as the most applicable intellectual property protection available to computer programs. This was due to the fact that computer programs and some of their related components tend to fit rather well into Copyright law. For example, the computer programs and their user manuals have been considered writings,[4] databases have *sometimes* been

---

[3] Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 430 (1984), *cited by* Justice Walker in Computer Assoc. International, Inc. v. Altai, Inc., 982 F.2d 693, 696 (2d Cir. 1992).

[4] *See* Synercom Technology v. University Computing Co., 462 F. Supp. 1003 (N.D.Tex.1978).

considered copyrightable compilations,[5] and the visual displays on computer monitors can be considered graphic works.[6] Also, the source code of a computer program, or the collection of statements or declarations which allows the computer programmer to communicate with the computer using a reserved number of instructions,[7] and the object code, or the representation of code that a compiler or assembler generates by processing a source code file,[8] are both copyrightable subject matter.[9]

By way of background, the Copyright Office began accepting copyright applications for computer programs in 1964.[10] As computers posed a new and intricate area of technology, Congress established the National Commission on New Technological Uses of Copyrighted Works (CONTU) in 1974.[11] The purpose of the CONTU was to study the implications of new and recent technologies, and to suggest that federal intellectual property law be revised to accommodate these changes.[12] After significant hearings and research conducted by experts, the board comprising CONTU decided that "it was clearly the intent of Congress to include computer programs within the scope of copyrightable subject matter in the Act of 1976."[13]

---

[5] *But see* Feist v. Rural Tel., 499 U.S. 340, (1991), and Bellsouth Adver. & Publ'g Corp. v. Donnelly Info. Publ'g, Inc., 933 F.2d 952 (11th Cir.1991).

[6] David W. Carstens, *Legal Protection of Computer Software: Patents, Copyrights, and Trade Secrets*, 20 J. CONTEMP. L. 13, 45 (1994).

[7] *See* Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521 (D. Colo.1992).

[8] *See* GCA Corp. v. Chance, C-82-1063-MHP, 1982 WL 1281 (N.D. Cal. Aug 31,1982).

[9] Carstens, *supra* note 6; *see also* 17 U.S.C. §106 (1988).

[10] PAUL GOLDSTEIN, COPYRIGHT, PATENT, TRADEMARK AND RELATED STATES DOCTRINES: CASES AND MATERIALS ON THE LAW OF INTELLECTUAL PROPERTY 846 (REV. 5TH ED., 2004) (1973).

[11] *See* National Commission on New Technological Uses of Copyrighted Works, *Final Report* (1978) [hereinafter CONTU]; *see also* Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41STANFORD L. REV.. 1045, 1046 (1989).

[12] Menell, *supra* note 11, at 1046.

[13] CONTU, *supra* note 11, at 16.

More significantly, the 1980 amendments to the 1976 Copyright Act[14] helped bring the Act up to speed with advances in technology,[15] including computer programs. The amended Act defined a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." [16]

The inclusion of such a definition in section 101 of the Copyright Act is an indication that Congress may have intended computer programs to be analyzed as copyrightable pursuant to that definition.[17] The problem with this strict definition is that it excludes from copyright manuals, flow charts, and any other representation that cannot be used to bring about that certain result.[18] However, in *Apple Computer, Inc. v. Franklin Computer Corp.*,[19] the Third Circuit held that so long as the computer program is original and fixed in a tangible medium, it will be copyrightable subject matter "whether it is expressed in words, in a flow chart, in source code or object code, and whether it is embodied in paper, magnetic disk tape or semiconductor chip."[20] Thus, the courts broadened the scope of the "fixed" requirement, opening the door for computer programs to become copyrightable subject matter.

But what does "fixed" actually imply? Section 101 of the Copyright Act sheds some light, and reads "[A] work is fixed in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced or otherwise communicated for a

---

[14] Copyright Act of 1976, 90 Stat. 2544 (1976) (codified as amended 17 U.S.C. § 101 (1980)).

[15] BERNARD A. GALLER, SOFTWARE AND INTELLECTUAL PROPERTY PROTECTION: COPYRIGHT AND PATENT ISSUES FOR COMPUTER AND LEGAL PROFESSIONALS 3 (Foreword by Jack E. Brown, 1995).

[16] GOLDSTEIN, *supra* note 10, at 77.

[17] William F. Patry, *Copyright and Computer Programs: It's all in the Definition*,14 CARDOZO ARTS & ENT. L.J. 1, 23 (1996).

[18] *Id.* at 32.

[19] Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, (3d Cir.1983), *cert. dismissed*, 464 U.S. 1033 (1984).

[20] *Apple Computer*, 714 F.2d at 1249

period of more than transitory duration."[21] Section 101's definition of "fixed" continues, "[A] work consisting of sounds, images, or both, that are being transmitted, is "fixed" for purposes of this title if a fixation of the work is being made simultaneously with its transmission."[22] The latter half of this definition would seem to be applicable to computer programs, and the Third Circuit's decision in *Williams Electronics, Inc. v. Artic International, Inc.*[23] discusses this requirement, "fixed in any tangible medium of expression," as it applies to video games.

In *Williams*, the court was presented with a dispute over two coin-operated video games, "Defender" manufactured by Williams Electronics, and "Defense Commander" manufactured by Artic International. Williams' game, "Defender," comprised several unique and original audio-visual features.[24] Artic's video game, "Defense Commander," was virtually identical to Williams'.[25] As a result, Williams brought suit against Artic alleging infringement of three of its "Defender" copyrights. The District Court agreed with Williams that Artic had in fact infringed three of its copyrights.[26] Artic appealed the lower courts judgment. The issue facing the appellate court was whether the audio-visual copyrights of "Defender" satisfied the "fixed in any tangible medium of expression" requirement.[27]

The Court of Appeals for the Third Circuit affirmed the lower court by holding that the audio-visual features of the "Defender" video game were fixed in a tangible medium of expression because there was always a repetitive sequence of

---

[21] 17 U.S.C. § 101 (1976).

[22] *Id.*

[23] Williams Elec., Inc. v. Artic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982).

[24] *Id*. at 872.

[25] *Id*. at 872–73.

[26] These acts of infringement included the infringement of the computer program copyright for the "Defender" game by selling kits which contained a copy of the video game; and infringement of two of Williams' audiovisual copyrights for the video game by selling copies of the game "Defender." *Id*. at 873.

[27] *See id*. at 873–74; *see also* GALLER, *supra* note 15, at 48.

a substantial amount of the sounds and images of the game, of which remained unchanged and constant regardless of how the player operated the game.[28] The Third Ciruit also found in favor of Williams on the object code issue and rejected Artic's argument that object codes cannot be protected because a copy must be intelligible to human beings and is only intended as a means of communication to humans.[29] The Third Circuit held, in response, that "[A] 'copy' is defined to include a material object in which a work is fixed 'by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.'"[30] [31]

Thus, in summary, the *Williams* case held that computer programs, which display on screen images, (like a vast majority of software on the market) are copyrightable because they are "fixed" in a "tangible medium"[32] as required by the Copyright Act. [33]

---

[28] *See Williams*, 685 F.2d at 874; *see also* GALLER, *supra*.

[29] *Williams*,. 685 F.2d at 876–77; *see also* GALLER, *supra*.

[30] *Williams*, 685 F.2d at 877; *see also* 17 U.S.C. §101 (1976).

[31] The *Williams* case came after a slew of earlier cases involving similar copyrighted video games and their infringement. *See e.g.*, Atari, Inc. v. North American Philips Consumer Electronics Corp., 672 F.2d 607 (7th Cir. 1982) *partly superseded by statute* as stated in Scandia Down Corp. v. Eroquilt, Inc., 772 F.2d 1423, (7th Cir. Ill. 1985) (the Atari court found for infringement and held that even though the defendant's game was not "virtually identical" to Pac-Man, it captured the "total concept and feel" of and was substantially similar to Pac-Man); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852 (2d Cir., 1982) (which held that the player's participation did not withdraw the audiovisual work from copyright eligibility); and Midway Mfg. Co. v. Dirkschneider, 543 F. Supp. 466 (D. Neb. 1981)(which held that the fact that the audiovisual works [video game] could not be viewed without a machine did not mean the works were not fixed).

[32] The tangible medium used here was the computer monitor, which, as the court held, was the equivalent of a machine.

[33] It is important to note that several aspects of the Artic game "Defense Commander" indicated that it was practically copied verbatim from Williams' game, "Defender." For example, an identical error of an earlier version of "Defender" showed up within the "Defense Commander" game at the exact same time it showed up in "Defender." *Williams*,. 685 F.2d at 876, n.6. In addition, and quite amusingly, both

Two other landmark decisions which help exemplify why computer programs are considered copyrightable subject matter include *Whelan Associates v. Jaslow Dental Laboratory, Inc.*,[34] and *Computer Associates International, Inc. v. Altai, Inc.*[35]

Whelan involved a dispute over a computer program which was designed to aid in the management of dental laboratories.[36] Whelan alleged that Dentcom's licensing of the Dentalab program, and Dentcom's program itself, infringed Whelan Associates' copyright in Dentalab.[37] The District Court found for Whelan on all claims.[38] The District Court held that Elaine Whelan was the sole author of the Dentalab program and that the agreement between Jaslow and Whelan made it apparent that the parties intended ownership

---

games displayed a listing of high scores achieved by past players by listing their initials, and the Artic game listed the initials of employees of Williams who achieved those scores on the "Defender" game. *Id*. at 876, n.6. Finally, Williams inputted a "hidden" copyright notice deep inside its memory devices and when the contents of Artic's memory were pulled up, so did Williams' "hidden" copyright notice. *Id*.

[34] Whelan Assoc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

[35] Computer Assoc. Int'l., Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992), *affirmed by* 1996 U.S. App. LEXIS 6363 (2d Cir. N.Y. Apr. 3, 1996), *cert. denied* 523 U.S. 1106 (1998).

[36] By way of background, the computer program at issue was called "Dentalab" and was written by an employee, Ms. Elaine Whelan, of an outside company, Strohl Systems Group, Inc. "Dentalab" was written in a computer program language called Event Driven Language (EDL). The program was successful and began operation at Jaslow in March of 1979. *Whelan*,797 F.2d at 1225–26. Ms. Whelan left Strohl shortly after developing Dentalab and formed her own company, Whelan Associates, Inc. *Id*. at 1226. After the formation of Whelan Associates, Jaslow entered into a written agreement with Whelan for the exploitation of the "Dentalab" program and to share in the profits. *Id*. Without informing Whelan of his plan, Rand Jaslow of Jaslow Labs, decided that the Dentalab program would benefit all other dental labs, like Jaslow, and began developing Dentalab in BASIC, as opposed to EDL, language. *Id*. [Many smaller dental labs computer systems did not operate in EDL language]. Rand then formed his own company, Dentcom, and sought to terminate his agreement with Whelan. It was this program that was the alleged infringer of Whelan's EDL Dentalab program. *Id.*

[37] *Whelan*, 797 F.2d. at 1227.

[38] *Id*. at 1228.

to lie solely with Whelan.[39] The court also determined that Whelan's copyright in Dentalab was in fact a valid copyright, and Dentcom's twenty-three sales of the Dentalab computer program were violations[40] of this copyright.[41] Finally, the district court held that the Dentalab copyright was infringed because the "overall structure" of Dentcom was substantially similar to the overall structure of Dentalab.[42]

On appeal, the Third Circuit adopted the substantial similarity test,[43] hinted at by the district court for determining copyright infringement, and found for Whelan. The court, relying on *SAS Institute, Inc. v. S & H Computer Systems, Inc.*,[44] held that the non-literal elements, or structure of a computer program, which Whelan alleged were infringed, were protectable elements under the Copyright Act.[45] The court determined that 17 U.S.C. § 102(a)(1) "extends copyright protection to 'literary works,' and computer programs are classified as literary works for the purposes of

---

[39] *Id*.

[40] To present a prima facie case of copyright infringement, the plaintiff must prove two things: 1) that they do in fact own the copyright which they allege to have been infringed, and 2) that the copyright has been copied by the alleged infringer. *See Id.* at 1231, *see also* Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp., 562 F.2d 1157, 1162 (9th Cir. 1977); Reyher v. Children's Television Workshop, 533 F.2d 87, 90 (2d Cir.), *cert. denied*, 429 U.S. 980 (1976).

[41] Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d, 1222, 1228 (3d Cir. 1986).

[42] *Id*. at 1233.

[43] The Substantial Similarity test is a single inquiry test where a finder of fact makes two findings of substantial similarity to support a copyright violation. First, the fact-finder must determine, with the aid of expert testimony, whether there is sufficient similarity between the two copyrighted works at issue to conclude whether the alleged infringer used the copyrighted work to make his/her own. Second, if the answer to the first question is in the affirmative, the fact-finder must determine, through layperson testimony, whether the copying was "illicit" or "an unlawful appropriation" of the copyrighted work. *See e.g.*, Arnstein v. Porter, 154 F.2d 464, 468–69 (2d Cir. 1946).

[44] SAS Institute, Inc. v. S & H Computer Systems, Inc., 605 F. Supp. 816 (M.D. Tenn. 1985) (the only other case, at the time, which addressed the issue of non-literal vs. literal elements of a computer program)

[45] *See Whelan*, 797 F.2d at 1238–39.

copyright."[46] The court also noted that the copyrights of other literary works can be infringed even when there is no substantial similarity between the works literal elements.[47]

The Third Circuit in *Whelan* created some controversy with its decision because it extended copyright protection to a facet of a program that lacked originality and completely ignored the idea-expression dichotomy.[48] However, the Second Circuit in *Computer Associates* abandoned the Third Circuit's reasoning in *Whelan.*[49]

The Second Circuit, on appeal from the Eastern District of New York, took one giant stride further in *Computer Associates*[50] than the Third Circuit did in *Whelan*. The court

---

[46] *Id.* at 1234; *see also* H.R. Rep. No.94-1476 at 5 (1976), 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S. Code Cong. & Ad. News 5659, 5667.

[47] Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d, 1222,1234 (3d Cir. 1986).

[48] Patry, *supra* note 17, at 51.

The court in *Whelan* made note that its decision would be put at odds with Justice Higginbotham's opinion in Synercom Tech., Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D. Tex. 1978), where Justice Higginbotham held that "input formats" of computer programs were ideas and not expression and, thus, were not copyrightable subject matter. Justice Becker noted that the two cases may be distinguishable but still proceeded to "come to grips" with Higginbotham's opinion in *Synercom*, and ultimately decided that Higginbotham's distinction between the copyrightability of sequence and form in the computer context and in any other context, was an incorrect one. *Whelan*, 797 F.2d at 1240. Becker's reasoning was that since Congress has not yet passed any special provision differentiating between ordering and sequencing of computer programs, Higginbotham was incorrect to take it into his own hands. *Id*.

[49] Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992). It is important to clarify that *Whelan* was decided by the Third Circuit and *Computer Associate's* was decided by the Second Circuit. Currently, the Supreme Court has yet to make a final determination as to the correct course of action. As it stands now, there is a split between the nine circuits, the majority of which follow the Second Circuit in *Computer Associates*, with a small minority following the Third Circuit's approach in *Whelan*.

[50] By way of background, both parties, Computer Associates [hereinafter CA] and Altai, were in the business of developing various types of computer software. Involved in this dispute was CA's program called "Scheduler," and Altai's program called "Zeke." "Scheduler" is a

devised a three prong "abstraction-filtration-comparison" test,[51] when deciding the issue of computer program copyright infringement. This test requires district courts to first dissect each component of the program and isolate each abstracted level contained in the specific computer program.[52] The court must then examine the structural components at each level that has been abstracted to determine if each abstraction is an idea or a copyright protected expression.[53] If the court found that any or all of the abstractions contained an expression, the final step was to determine if any of the expressions have been copied by the defendant.[54]

---

program designed for IBM computers which acts as a business scheduler, and contains an important sub-component called "Adapter" which acts an operating system compatibility component, or translator. "Scheduler" was designed to operate on an IBM System 370 family of computers which contain one of three operating systems, DOS/VSE, MVS, or CMS. "Adapter" is what allows "Scheduler" to operate on multiple operating systems without having to change software. Altai's "Zeke" program was designed to operate on a MVS operating system, similar to "Scheduler." However, before "Zeke" became operational on a MVS system, an employee of Altai, James P. Williams, approached Claude F. Arney, III, a programmer employed by CA, about the possibility of him coming to Altai to help develop "Zeke's" MVS system compatibility. Arney participated substantially in the creation of "Adapter" while he was employed at CA, and although Williams knew about "Scheduler" and "Adapter," he did not know that "Adapter" was a component of "Scheduler." Arney left CA to work for Altai shortly thereafter and took copies of the source code for both the VSE and MVS versions of "Adapter." Arney then, using the "Adapter's" source code, created two versions of "Oscar," Altai's version of "Adapter," which permitted "Zeke" to operate on a MVS and VSE operating system.

[51] Computer Assocs. Int'l., Inc v. Altai, Inc., 982 F.2d 693, 706 (2d Cir. 1992) ; *see also* Justice Learned Hands decision in Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir.1930) (where Justice Hand articulated what is now known as the "abstractions" test for separating an idea from an expression [hereinafter referred to as the Idea-Expression dichotomy].

[52] *Computer Assocs.*, 982 F.2d at 706–07.

[53] *Id.* at 707; *see also* 3 NIMMER ON COPYRIGHT, § 13.03[F], at 13 62.34-63 (discusses the "successive filtering method" adopted by Justice Walker's "filtration" prong to the "abstraction-filtration-comparison" test used in *Computer Associates*).

[54] *Computer Assocs.*, 982 F.2d at 710.

This approach from *Computer Associates* has been said to be overly complicated, while the *Whelan* approach has been said to be overly simplistic.[55] Which approach is most effective? The three-prong test from *Computer_Associates*. This is chiefly because it gives the court the opportunity to break down the program to its "bare-bones" to see what *actually* comprises it, which arguably, would allow the court to make a sound, correct decision when faced with a copyright infringement suit involving a computer program. The downside to this approach, well presented by Professor Patry,[56] is that the three prong test devised by the Second Circuit benefits only one type of person, the expert witness, whose services are desperately needed by the courts since most judges, though not *all*, do not have backgrounds in computer programming.[57]

The preceding discussion has focused on the history and reasonably simplistic nature of computer program protection under United States copyright law.[58] The next section turns

---

[55] Patry, *supra* note 17, at 55

[56] Patry, *supra* note 17, at 55.

[57] Remember, it is imperative for attorneys and law students to pay very close attention to the jurisdiction in which they are located to determine which approach the court in that jurisdiction follows. Remember, there is a split among the circuits as to which test is applied, i.e. the substantial similarity test devised by the Third Circuit, and the abstraction-filtration-comparison test developed by the Second Circuit.

[58] For an example of computer programs that have purposes to commit copyright infringement, *see* BMG Music, et al. v. Gonzalez, 430 F.3d 888 (7th Cir. 2005). (defendant was found liable, at summary judgment, for "direct copyright infringement" by downloading 30 copyrighted songs onto her computer using an on-line computer program search engine. The court, in finding for the plaintiff recording companies, rejected the defendants "fair use" and "innocent infringer" defenses. *See also In Re* Aimster Copyright Litig., 334 F.3d 643 (7th Cir. 2003) (it is not enough that a product [computer software] be capable of non-infringing uses. A defendant must show evidence that the service has *ever* been used for such non-infringing uses); and A&M Records Inc. v. Napster, 239 F.3d 1004 (9th Cir. 2001) (on its face the Copyright Act's definition of digital audio recording devices, computers are not digital audio recording devices because their 'primary purpose' is not to produce digital audio copied recordings).

to computer program protection under United States patent law.

B. United States Patent Law as Applied to Computer Programs

> "A patent offers greater protection . . . than a copyright, for it can protect the principle underlying a mechanism or process as well as the specific form." [59]

Patents are a very powerful and desirable protection for inventors.[60] As such, a computer program patent would make it illegal for any unauthorized person to make, use, or sell the patented program.[61] However, it is extremely difficult for patent applicants to achieve that level of security because the United States Patent and Trademark Office ("PTO") has treated applications for computer programs as a "special case, and set hurdle after hurdle in the way of treating them as patentable subject matter under section 101 of the Patent Act."[62] Some have even claimed that software patents are the most scrutinized within the intellectual property world.[63] The problems with computer programs are that they do not easily meet the strict standards for patents – novelty, utility and non-obviousness – and the Supreme Court has made it clear that these tests set a high standard.[64]

Someone new to patent law may first look at computer programs and see a simple process, which, under the Patent

---

[59] Jean F. Rydstrom, *Patentability of Computer Programs*, 6 ALR Fed 156, 160 (1971), *citing* Bender, *Computer Programs: Should They be Patentable*? 68 COLUMBIA L. REV. 241 (1968).

[60] GALLER, *supra* note 15, at 31.

[61] *Computer Programs and Proposed Revisions of the Patent and Copyright Laws*, 81 HARVARD L. REV. 1541, 1553 (1968).

[62] GOLDSTEIN, *supra* note 10, at 952.

[63] *See e.g.* Anthony E. Anderson, Article, *Taming The Code: Effectively Implementing Software Patents*, 5 J. MARSHALL REV. INTELL. PROP. L. 381, 382 (2006).

[64] Computer Programs, *supra* note 62; *citing* Graham v. John Deere, Co., 383 U.S. 1, 19 (1966).

Act, would be patentable subject matter.[65] However, while simple, these processes can create problems for the courts, like ones with limited capabilities such as a process which recites mathematical algorithms, and, according to the courts, such mathematical algorithms were once not viewed as patentable subject matter.[66] The issues concerning the patentability of computer programs has been a hot topic of judicial debate over the past twenty-five or so years, and this debate among the courts began with the 1972 Supreme Court decision, *Gottschalk v. Benson*.[67]

The Supreme Court in *Benson* was faced with deciding whether an invention described as being similar to "the process of data by program and more particularly to the programmed conversion of numerical information" in general-purpose digital computers.[68] To simplify the above invention, this process was essentially a method for "converting binary-coded decimal (BCD) numerals into pure binary numerals."[69] This method was what the court ultimately coined as a mathematical algorithm, or a "procedure for solving a given type of mathematical problem."[70]

Procedurally, in *Benson*, the PTO examiner initially rejected two claims of the application, claim 8 and claim 13, as being outside the scope of section 101 of the Patent Act.[71] The Court of Customs and Patent Appeals, however, sustained the same two claims which the PTO initially rejected.[72] On appeal, the Supreme Court agreed with the PTO examiner, and held that this algorithm was outside the scope of patentable subject matter pursuant to section 101 of

---

[65] Carstens, *supra* note 6, at 20.

[66] *Id*.

[67] Gottschalk v. Benson, 409 U.S. 63 (1972).

[68] *Id.* at 64; *see also* Ralph D. Clifford, *The Federal Circuit's Cruise to Uncharted Waters: How Patent Protection for Algorithms and Business Methods May Sink the UCITA and State Intellectual Property Protection*, 73 TEMPLE L. Rev. 1241, 1243 (2000), and Carstens, *supra* note 6

[69] *See Benson*, 409 U.S. at 64.

[70] *Id*. at 65.

[71] *See Id*., *see also* Clifford, *supra* note 68

[72] *Benson*, 409 U.S. at 65

the Patent Act. In so holding, the Court stated that "the mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that . . . the patent would wholly preempt the mathematical formula and . . . would be a patent on the algorithm itself."[73] Essentially what the Court did in *Benson* was create an exception to the scope of section 101 of the Patent Act, known as the "mathematical algorithm" exception,[74] and refused to extend patent protection to computer programs because the algorithm "had no substantial practical application."[75]

Despite its admitted, albeit disguised, ignorance on the subject,[76] the Supreme Court, upheld its *Benson* decision in the subsequent case of *Parker v. Flook.*[77] This case involved an application for a patent on a "Method of Updating Alarm Limits."[78] The Court made an important initial distinction of this method in noting that the only novel feature of the method was the mathematical algorithm.[79]

---

[73] *Id.* at 71–72.

[74] *See* Clifford, *supra* note 68 at 1242.

[75] *Benson*, 409 U.S. at 71.

[76] Of important note to this discussion, is that the Supreme Court, in dicta, made a few statements that seemed to be an "ill-fit" with the crux of its argument. For example, the Court stated first that "[I]t may be that the patent laws should be extended to cover these programs, a policy matter to which we are not *competent* to speak *[emphasis added]*. *Benson*, 409 U.S., at 257. The court then went on and practically conceded its ignorance on the topic when it said "[I]f these programs are to be patentable, considerable problems are raised which only committees of Congress can manage . . ." *Id*., at 258. In its argument, the Court seemed to proffer a fairly coherent opinion while rejecting the patentability of a mathematical algorithm, but at the same time it conceded that it was totally ignorant on the subject. So it is in this writers opinion that the Supreme Court was not yet prepared to handle such a case, and truly find it difficult to believe that this opinion was afforded a great amount of relevance.

[77] Parker v. Flook, 437 U.S. 584, 585 (1978).

[78] *Id*. at 585.

Essentially, an "alarm limit" is a number. These "alarm limits" were to be used in conjunction with catalytic conversion processes where the "alarm limit" would alert the user when the process approached an abnormal condition.

[79] *Id.* at 585.

Procedurally, the PTO examiner once again rejected the application and found that "the mathematical formula constituted the only difference between the prior art[80] and the invention."[81] Following its actions taken in the Flook litigation, the C.C.P.A (predecessor to the Court of Appeals for the Federal Circuit) reversed the PTO by reading *Benson* to apply to only those claims which sought to entirely preempt a mathematical algorithm, and that since the inventor's process would not constitute infringement of the claims, a patent would not preempt the algorithm.[82] Also adhering to its precedent was the Supreme Court, which reversed the C.C.P.A., and held that "[R]espondent's application simply provides a new and presumably better method for calculating alarm limit values."[83] The Court opined further, and more importantly, that the " . . . process is un-patentable under § 101, not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention."[84]

In *Flook*, the Supreme Court once again rejected the patentability of a mathematical algorithm/computer program, but the *Flook* decision was not unanimous, as there were three dissenting Justices. The crux of the Dissent's argument was that this case was far different from *Benson* in that the mathematical algorithm at issue in *Flook* comprised only one step of the entire process, whereas in *Benson*, the mathematical algorithm *was* the invention.[85] The Supreme Court's stubbornness to uphold a patent for a "computer

---

[80] Prior Art is the crux of the statutory bar of novelty under Patent law. An invention is in the prior art if it has already been patented or described in a printed publication in any country, or if it is in public use or on sale in the United States, for more than one year prior to the date the patent is applied for. *See generally* 35 U.S.C.A. § 102(b).

[81] *Flook*, 437 U.S. at 587.

[82] *Id*.

[83] *Id*. at 594–95.

[84] *Id*. at 594.

[85] *See id*. at 599.

related" program did not stop the Federal Circuit from holding otherwise, and the debate continued.

For example, the Federal Circuit's decision, *Application of Diehr*,[86] demonstrated its stern opposition to the Supreme Court's view on the subject.[87] Reminiscent of its previous actions, the PTO examiner, utilizing the Court's reasoning from *Flook*, rejected the inventor's claims[88] On appeal, however, the Federal Circuit held that the Supreme Court's reasoning in *Benson* and *Flook* did not deny patentability to a claim as non-statutory simply because it involved a computer program or because it was computer related.[89] The Federal Circuit was of the opinion that "claims may be rejected under section 101 because they attempt to embrace *only* . . . a mathematical algorithm . . . but not merely because they define inventions having something to do with a computer."[90]

Basically, instead of removing the algorithm claim to determine the patentability of the remaining claims [like the Supreme Court], the Federal Circuit in *Diehr* looked at *all* the claims, including the algorithm claim, and found for patentability because the claims, as a whole, satisfied the requirements for patent. Once again, the Supreme Court granted certiorari to determine if this court was correct.[91]

Astonishingly, the Supreme Court upheld the Federal Circuit's granting of the patent at issue, marking the first time the "high-nine" affirmed the patentability of a computer-related invention.[92] The Supreme Court held the way it did because it totally set aside anything to do with the mathematical algorithm and decided whether the process for molding rubber, *as a whole*, was patentable subject matter

---

[86] Application of Diehr, 602 F.2d 982 (1979).

[87] *Diehr* involved a method for operating molding presses used during the manufacture of rubber products. This method, to be successful, relied upon a mathematical equation called the "Arrhenius equation." For a discussion on the Arrhenius equation, *see* Diamond v. Diehr, 450 U.S. 175, 178, n. 2 (1981).

[88] *Application of Diehr*, 602 F.2d at 984–85.

[89] *Id.* at 986.

[90] *Id.* at 986–87.

[91] *See Diamond*, 450 U.S. at 175.

[92] *See Id*.

within section 101 of the Patent Act.[93] This landmark decision, although very similar to the *Benson-Flook* line of cases, opened the floodgates for computer program patents in the United States.[94]

Subsequent to the court's decision in *In Re Lowry*,[95] the Patent and Trademark Office began to take notice of the issue and realized that it was time to start paying heed to patent applications for computer-type programs. In *Lowry*,[96] the court noted that in determining the novelty of a computer memory, which embodied a novel data structure, the Patent and Trademark Office could not disregard the data structure on the ground that it constituted unpatentable printed matter.[97] In its decision, the Federal Circuit essentially summoned the PTO, and for the PTO, it was a wake-up call.

---

[93] *Id*. at 191.

[94] Following the *Diehr* decision, a vast amount of Federal Circuit cases upheld patents for computer-related programs in 1994, including In Re Alappat, 33 F.3d 1526 (Fed.Cir.1994) (where it was held that a patent that functionally defined a machine that manipulated data for an oscilloscope display was patentable subject matter under §101 of the Patent Act); In Re Warmerdam, 33 F.3d 1354 (Fed.Cir.1994) (which held that a machine with a data structure embedded in its memory qualified for a patent, although a method for generating a data structure did not); and In Re Schrader, 22 F.3d 290 (Fed.Cir.1994) (where the court held that when a functional step [i.e., a physical display of data] was added to a once unpatentable algorithm, the whole method would pass muster for patentability); *see also* Goldstein, *supra* note 10, at 953.

[95] In Re Lowry, 32 F.3d 1579 (Fed.Cir.1994); *see also* GOLDSTEIN, *supra* note 10, at 953.

[96] *Lowry* involved a patent application for a "Data Processing System Having a Data Structure with a Single, Simple Primitive." 32 F.3d at 1580. This invention was basically a memory storage program and effectively organized and stored data in a computer memory. The Patent examiner rejected claims 1-5 of the application as unpatentable subject matter pursuant to the section 101 of the Patent Act, and also rejected claims 20–29 as unpatentable pursuant to section 102(e). Claims 1–19 were also rejected pursuant to section 103 of the Act. The Board reversed the Examiners rejection of claims 1–5 and held that "a memory containing stored information, as a whole, recited an article of manufacture," and was patentable subject matter. *Id*. at 1582. The Board affirmed the Examiner's rejection pertaining to sections 102(2) and 103. Lowry appealed.

[97] *In Re Lowry*, 32 F.3d at 1582.

Subsequent to *Lowry*, yet another controversial, but important case in the computer program patent war came down from the United States Court of Appeals for the Federal Circuit in *State Street Bank & Trust Co. v. Signature Financial Group, Inc*,[98] where the court expressly ignored the Supreme Court's "mathematical algorithm exception."

In *State Street*, a patent for a "data processing system for Hub and Spoke Financial Services Configuration"[99] was applied for.[100] The court was forced to decide whether this patent was invalid for failing to state a claim of statutory subject matter as required pursuant to section 101 of the Patent Act. The court held, that the "mathematical algorithm exception," created by the Supreme Court in *Benson*, when applied to Claim 1 of the "hub and spoke" invention at issue, was not applicable because the algorithm encompassing Claim 1 produced a "useful, concrete, and tangible result," which "renders it statutory subject matter, even if the useful result is expressed in numbers . . ."[101] The court upheld the patent.[102]

State Street has been viewed by some to be a controversial decision because it upheld a patent that claimed nothing more than mathematical equations while at the same time failed to claim *any* specific algorithm.[103] Some have argued that the court upholds patents, like the one at issue in

---

[98] State Street Bank & Trust Co. v. Signature Fin. Group, 149 F.3d 1368 (1998).

[99] Essentially, this invention facilitated a structure whereby mutual funds pool their assets into an investment portfolio organized as a partnership. *See Id*. Such a system is referred to as a "Hub and Spoke" system. *See id*.

[100] *State Street*, 149 F.3d at 1370.

[101] *See id.* at 1375; *see also* In Re Alappat, 33 F.3d 1526 (Fed. Cir. 1994) (data transformed by a machine through a series of mathematical equations which produced a smooth display on a rasterizer monitor, constituted "a practical application of an abstract idea" because "a useful, concrete and tangible result" was produced).

[102] The Federal Circuit acted to essentially dispose of the so-called "business method exception." The business method exception was basically a judicially created exception to section 101 of the Patent Act, very similar to that of the mathematical algorithm exception.

[103] *See* Jay Dratler, Jr., *Alice in Wonderland Meets the U.S. Patent System*, 38 AKRON L. REV. 299, 302–03 (2005).

State Street, because the PTO and the Federal Circuit have developed a "land rush" mentality toward patents, meaning the more patents they can issue the better.[104] Such a mentality has been said to create a significant mental block among the basic institutions of this country's original patent system concerning the simple, but important, notation of balance, which had made this system so effective in the first place.[105]

Notwithstanding its controversial decision, the Federal Circuit subsequently decided *AT&T Corp. v. Excel Communications, Inc.*,[106] and effectively upheld its contentious decision in *State Street*, and again refused to apply the mathematical algorithm exception.

In *AT&T*, the court was faced with deciding whether an invention described as a "Call Message Recording for Telephone Systems" was patentable subject matter within the meaning of section 101 of the Patent Act.[107] The PTO initially rejected this patent application for reasons unrelated to section 101.[108] The district court upheld the PTO's rejection on the grounds that the patent implicitly recited a mathematical algorithm and that the only physical step in the claims involved the data-gathering process for the algorithm, and thus, was excluded from patentability under the "mathematical algorithm exception."[109]

On appeal, the Federal Circuit noted, at the outset, that "[W]hether stated implicitly or explicitly, we consider the scope of section 101 to be the same regardless of the form – machine or process – in which a particular claim is

---

[104] *See id.* at 303–04.

[105] *See id.* at 304.

[106] AT&T Corp v. Excel Commc'ns, Inc., 172 F.3d 1352, (1999); *cert. denied* 528 U.S. 946 (1999).

[107] *See id.* at 1353.

This Call Message Recording System contained a three step process for when a caller made a direct-dialed long distance telephone call: 1) after the long-distance phone call was placed and recognized by its local exchange carrier (LEC) it would be routed automatically to a primary inter-exchange carrier (PIC); 2) the PIC then carries the phone call to the LEC which serves the recipient of the call; and 3) the recipient's LEC delivered the call to the recipient, using the local network. *See Id.* at 1354.

[108] *AT&T*, 172 F.3d at 1354.

[109] *Id*. at 1355–56.

drafted."[110] "In fact, whether the invention is a process or a machine is irrelevant . . . and we are comfortable in applying our reasoning in *Alappat* and *State Street* to the method claims at issue in this case."[111] Thus, the court expressly affirmed its decisions in those two previous cases. Thereafter, the Federal Circuit focused its analysis on the issue at hand, "whether the mathematical algorithm is applied in a practical manner to produce a useful result."[112] The court held in the affirmative. It held that the district court failed to apply the appropriate analysis to the method claims at issue and that had they conducted the proper analysis, it would have easily found that the claims applied for would have clearly fallen within the scope of section 101 of the Patent Act.[113]

The Federal Circuit has effectively discarded the "mathematical algorithm exception" once implemented by the Supreme Court in *Benson*, and has essentially permitted the patenting of both computer program patents, patents that comprise a mathematical algorithm, and also patents for business methods. As a result, the PTO has been showered with applications for the very same.

One can see the difficulty computer programs cause for the PTO and for the courts in this country. To help evade the eradication of the United States Patent system as applied to computer programs, this article now turns to a discussion on how international governments have attached intellectual property protection for computer programs.

II. COMPUTER PROGRAM PROTECTION UNDER AN INTERNATIONAL INTELLECTUAL PROPERTY LAW REGIME

The purpose of this section is to introduce a model of an international approach to intellectual property law as it

[110] *Id*. at 1357.
[111] *Id*. at 1358.
[112] *Id*. at 1360.
[113] *Id*. at 1361.

applies to computer programs. Although only the surface will be penetrated, this section aims to summarize the approach taken by the European Community in codifying its own intellectual property legislation specific to computer programs. European intellectual property law was chosen to assist the law student and/or attorney with this discussion because Europe follows an approach in which the United States should follow. Review of international approaches was also chosen because in the United States legal education system, exclusive of courses and programs devoted to international law, there is very little exposure to international legislation, particularly international intellectual property law.[114]

A. The European Community's Approach to Intellectual Property Protection for Computer Programs

Research conducted on this issue, and how it has been approached by the European Community, revealed a very impressive approach as to how the European Community ("EC")[115] handled the complexity of the situation that is computer programs. On May 14, 1991, the EC's Council of Ministers adopted the Directive on the Legal Protection of Computer Programs.[116] With the enactment of this Software Directive, the EC leapt ahead of the United States and Japan, two world leaders in computer programming and technology.[117]

What is ideal about this directive is that its drafters recognized the dilemma concerning computer programs. In the Preamble of the Software Directive, for example, several

---

[114] This, of course, is also exclusive of courses and/or lectures about the Berne Convention on Copyright, and other similar treaties to which the United States is, or is not a party or signatory to.

[115] The European Community is now commonly referred to as the European Union, or EU.

[116] Council Directive 91/250/EEC of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J. (L 122) 42 [hereinafter Software Directive]; *see also* Palmer & Vinje, *supra* note 121, at 65.

[117] *See* Palmer & Vinje, *supra* note 121, at 66.

provisions explain why this Directive has been passed; one such provision reads, " . . . computer programs are playing an increasingly important role in a broad range of industries and computer program technology can accordingly be considered as being of fundamental importance for the Community's industrial development."[118] This provision clearly recognizes the significance of computer programs in European society. Article 1(1) continues, "[I]n accordance with the provisions of this Directive, Member States shall protect computer programs, by copyright, as literary works within the meaning of the Berne Convention…."[119] The EC has recognized the significance of computer programs, the problems associated with protecting them, and that they should be protected under European copyright law as literary works.

With the passing of the Software Directive, the EC has essentially created a *sui generis*[120] body of law, which is the approach that this writer suggests the United States Congress must take. [121]Several Articles of the directive mirror United States copyright law. Specifically, Article 4(a) of the Directive includes some very special language which the

---

[118] Software Directive, *supra* note 116, at Preamble.

[119] *Id.* at Article 1(1).

Also, Article 1(2) stated, similar to the American Idea-Expression dichotomy, that only the expressions of a computer program are to be protected by this Directive and any ideas and principles associated with any element of a program are not to be protected under the Directive.

[120] For a discussion on *sui generis* legislation, *see* Pamela Samuelson, *Creating a New Kind of Intellectual Property: Applying the Lessons of the Chip Law to Computer Programs*, 70 MINN. L. REV. 471 (1986).

[121] Although the Software Directive specifically states that copyright law is to be used, the EC has devised legislation specific to computer programs, which comprises some copyright and some patent law in terms of the American definition, which is very much like creating a *sui generis* body of law. Congress has enacted *sui generis* legislation before when it passed the Semiconductor Chip Protection Act of 1984, Pub. L. No. 98-620, tit. III, 98 Stat. 3347 (codified at 17 U.S.C. §§901-914 (Supp. II 1984)); *see also* Samuelson, *supra* at 471 n.3. Prior to that Act, Congress had enacted *sui generis* type legislation when it provided for protection for non-obvious ornamental designs for articles of manufacture in 1870, *see* Act of July 8, 1870, ch. 230,§71-76, 16 Stat. 198, 209–10 (codified at 35 U.S.C. §171-173 (1982)) *cited by* Samuelson, *supra* at 472 n.4.

United States must implement into its own *sui generis* legislation.

Article 4(a) of the Directive permits the owner of a program to have the exclusive right to do or authorize ". . . the permanent or temporary reproduction of a computer program by any means and in any form, in part or in whole."[122] This sounds very similar to United States copyright law. Article 4(a) continues, "[i]nsofar as loading, displaying, running, transmission or storage of the computer program necessitate such reproduction, such acts shall be subject to authorization by the rightholder."[123] Again, this parallels United States Copyright law. This section of the Directive reads like a hybrid body of law comprising mostly copyright law, with some patent law mixed in. The first part of Article 4(a) reads much like the copyright system in the U.S., giving the holder the absolute right to copy his or her work, and the second part reads somewhat like patent law giving the holder the exclusive right to control how the program is used, reproduced or distributed, including the holder's right to express authorization. However, to better understand what Europe has actually done, let us explore Article 4 a little more deeply.

Article 4(b) of the Software Directive permits the right holder with "the right to do or to authorize . . . the translation, adaptation, arrangement and any other alteration of a computer program and the reproduction of the results thereof . . ."[124] Article 4(c) permits the right holder "to do or to authorize. . . any form of distribution to the public . . . of the original computer program or of copies thereof."[125]

The remainder of Article 4 illustrated above seems to represent more copyright law than patent law. For example, Article 4(b) permits the author to control how the computer program is translated, adapted, arranged or altered, which is one of the protections granted under American copyright law. Article 4(c) of the Directive permits the author to control how

---

[122] Software Directive, *supra* note 124, at art. 4(a).
[123] *Id.*
[124] *Id.* at Art. 4(b).
[125] *Id.* at Art. 4(c).

the computer program is distributed to the public, which is also a protection granted under American copyright law. The right to use, make or sell an invention[126] is the crux of American patent law, and some of that can be read into Article 4(c) above.[127] So, although Article 4(a) and 4(c) contain some language which can be associated with American patent law, Europe has essentially taken European copyright law and explicitly applied it to computer programs.[128]

In so doing, Europe has created a unique and peculiar body of law, *sui generis,* by taking bits and pieces of its own Copyright law, and applied it directly to computer programs. Undoubtedly, it would not seem to be all that taxing for Congress to take a similar approach and enact a *sui generis* body of law,[129] both detailed and specific, which would act to

---

[126] *See generally* 35 U.S.C.A §154.

[127] For example, the phrase "distributed to the public," may be interpreted to possibly mean "selling."

[128] For an analysis of the Software Directive in the European Court of Justice, *see* Order of the President of the Court of First Instance of 22 December 2004, Microsoft Corp. v. Comm'n of the Eur. Cmtys, *Proceedings for interim relief* - Article 82 EC. Case T-201/04 R, *available at* 2004 ECJ CELEX LEXIS 704 (2004) (this case was a dispute between Microsoft Corporation and Sun Microsystems, where Sun alleged that Microsoft was abusing its dominant position when it refused to disclose to it the technology necessary to allow interoperability of its work group server operating system with the Windows Client PC operating system. The court noted that the communications protocols were protected by copyright under the Berne and by Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs. Thus, the court held that although Microsoft's communications protocols were protected by copyright, implementation of the communications protocols does not constitute a form of exploitation prohibited by copyright).

[129] Congress has already enacted quasi-*sui generis* legislation when it extended copyright protection to computer programs in machine-readable form when it passed section 10 of the Act of December 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015, 3028 (codified at 17 U.S.C. §101 (1982) and 17 U.S.C.A. §117, *cited by* Samuelson, *supra* at 474 n. 12. However, these amendments made to the Copyright Act were relatively minor, because all Congress did was add a definition of computer program into section 101, and by substituting an amended section 117 which gave

simplify intellectual property protection for computer programs in the United States system.

III. WHAT NEEDS TO BE DONE

Currently, the Patent and Trademark Office is teeming with patent applications, where approximately 443,652 patents were applied for in the 2006 fiscal year.[130] In the same year, only 183,187 patents were issued.[131] That means 260,465 patent applications were either denied or not prosecuted. With respect to patent applications for computer software, in 2003 there were approximately 25,475 patent applications for computer software and approximately 12,371 applications for mathematical algorithms.[132] In addition, an application for software will be pending, on average, for forty-four months.[133] These facts have created significant uproar amongst proponents of computer software patents.[134] Because of this alarming number of patent applications, and the amount of time it can take to prosecute a computer software patent, the Patent Act needs major revision.

Specifically, section 3 of the Patent Act[135] must be revised to address this patent mishandling problem by

---

owners of copyrighted computer programs a limited right to modify them and archive copies of them. *See e.g.*, Samuelson, *supra*, at 474–75 n. 12.

[130] United States Patent and Trademark Office, 2006 Fiscal Year USPTO Workloads, Table 2, available at http://www.uspto.gov/web/offices/com/annual/2006/50302_table2.html [Hereinafter USPTO] (This number is a 312,249 increase from the 131,403 patents applied for in 1986).

[131] *Id*. at Table 6.

[132] Allen Clark Zoracki, *When is an Algorithm Invented? The Need for a New Paradigm for Evaluating an Algoritgm for Intellectual Property Protection*, 15 Alb. L.J. Sci. & Tech. 579, 585 (2005). [I was unable to verify the accuracy of these numbers, hence why the term approximately was used. I was also unable, after an exhaustive search, to locate more current numbers. However, it is fair to say that the number of applications for software has drastically increased from 2003].

[133] USPTO, *supra* at Table 4.

[134] *See* Anderson, *supra* note 64, at 383.

[135] Section 3 of the United States Patent Act, 35 U.S.C.A. §§1-376, is the section that establishes the directors and employees of the PTO. I believe the scope of section 3 should be broadened and/or specified to

broadening the scope of examiners permitted to prosecute patents at the PTO. The only way computer program applications can be prosecuted fairly and thoroughly, is to increase the number of examiners, as a whole, and also ensure that they come with computer programming backgrounds.

In the alternative, if the Patent Act cannot undergo major revision, and pursuant to the goal and purpose of this article, Congress must enact a completely separate body of law specific to computer programs. Such a distinctive body of law is referred to as a *sui generis* legal regime, taking bits and pieces from both copyright and patent, and blending them together.[136] By passing such a body of law specific to computer programs, this country will be able to avoid future divergence between the courts, and would also ease the tension and stress of prosecuting computer program patents.

IV. CONCLUSION

Should Congress use copyright law, patent law or a quantity of both? You may recall this question being posed to you at the beginning of this note. That is the question that I have attempted to resolve throughout, and the answer to this question should have been readily apparent, *sui generis* legislation is needed.

As articulated above, *sui generis* legislation is a body of law that is unique or peculiar, and not specific to one area of law.[137] This is chiefly the case because one specific area of law is not, in itself, always sufficient to deal with a complex legal issue; here, intellectual property protection for computer programs.[138] In dealing with the convolution that is caused by computer programs, copyright law and patent law, respectively each by itself, do not offer a comprehensive

---

allow for examiners to come from a broader range of educational backgrounds, including computer programming.

[136] Palmer & Vinje, *The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development*, 2 DUKE J. OF COMP. & INT'L L. 65, 66 (1992).

[137] *See* BLACK'S LAW DICTIONARY, *supra* note 2.

[138] *Id*.

body of legislation. Such a lack of depth ultimately deprives the courts of the opportunity to meticulously preside over litigation involving computer programs. Thus, it is imperative for Congress to enact *sui generis* legislation specific to computer programs, whereby molding pieces of both copyright law and patent law together into one unique body of law.

To achieve such a result, Congress should mirror the Software Directive enacted by Parliament in the European Community. It is clear that the EC Parliament recognized the crisis that computer programs presented to its intellectual property legislation, and it wasted little time in enacting appropriate legislation to resolve that problem. Hopefully, Congress can work toward achieving the same goal