1993

Against a Sui Generis System of Intellectual Property for Computer Software

John M. Griem Jr.

Follow this and additional works at: http://scholarlycommons.law.hofstra.edu/hlr

Part of the Law Commons

Recommended Citation
Available at: http://scholarlycommons.law.hofstra.edu/hlr/vol22/iss1/4

This document is brought to you for free and open access by Scholarly Commons at Hofstra Law. It has been accepted for inclusion in Hofstra Law Review by an authorized administrator of Scholarly Commons at Hofstra Law. For more information, please contact lawcls@hofstra.edu.
NOTE

AGAINST A SUI GENERIS SYSTEM OF INTELLECTUAL PROPERTY FOR COMPUTER SOFTWARE

I. INTRODUCTION

Today, computer technology is an integral part of the worldwide economy. It plays an increasingly significant role in nearly every aspect of modern business life, from production to communications to design. As with any technology, designers and manufacturers have sought to secure reliable and enforceable intellectual property protection for their creations. Only this type of protection will enable the original designers to prevent others from appropriating the ideas and innovative effort which they have invested in creating their products.

In general, intellectual property protection for computer hardware, which consists essentially of complicated electrical apparatus, has been based exclusively and without controversy in patent law. Computer software, however, is a technology unlike any that society has


2. In addition, mask works of semiconductor chips can be separately protected. See discussion infra note 18.

3. Today, the instructions in software can be permanently imprinted as hardware and functions traditionally associated with hardware can be performed by software. Consequently, the term “computer software” is used in this Note in the broadest possible sense. The term encompasses any collection of coded instructions used to direct the overall functioning of peripheral machines connected to a central processing unit and any set of instructions which directs the internal functioning of the central unit itself, whether those instructions are permanently imprinted as hardware or not. As used here, the term includes programs which run on a general-purpose computer and perform only pure data-manipulation functions as well as programs which direct the progress of an industrial process. See OTA REPORT, supra note 1, at 125-30; Nelson R. Capes, Current Status of Patent Protection for Computer Software, 74 J. PAT. & TRADEMARK OFF. Soc’y 5, 5-9 (1992); see also Randall Davis, The Nature of Software and Its Consequences for Establishing and Evaluating Similarity, 5 SOFTWARE L.J. 299, 315 & n.17 (1992).
ever known. It has a unique dual nature, with both expressive and innovative aspects. Ever since the birth of an independent software industry, this dual nature has created considerable controversy as to the proper form and scope of the intellectual property protection which should be applied to it. Nearly all judicial authorities and academic commentators agree that copyright law is applicable to protect the most expressive aspects of widely distributed computer software. Neither judges nor academic commentators agree, however, on the proper extent of copyright protection for computer software, nor have they agreed on whether patent law should be applicable at all to computer software.

4. The birth of independent software industry can be considered the day in June of 1969 when International Business Machines announced that it was “ unbundling” software from the sale of hardware. At that time the practice of “ bundling” was under attack as a violation of antitrust laws. David Bender, The More Things Change, the More They Stay the Same: An Unhurried Reflection on Software Protection Over the Years, 16 Rutgers Computer & Tech. L.J. 309, 311 (1990) [hereinafter Bender, The More Things Change]; David Bender, The Renaissance of the “Software Patent”, 13 Hamline L. Rev. 205, 207 (1990) [hereinafter Bender, The Renaissance].

5. The focus of this Note is on widely distributed software, which is by far the largest proportion of software created. In contrast, trade secret protection has been used, and indeed continues to be used, to protect from theft both the expression and applied ideas in limited-distribution and custom-designed software under a system of contractual secrecy obligations. See Michael D. Stein, The Importance of a Trade Secret as a Supplement to Copyright Protection of Computer Software, IPL NewsL., Fall 1993, at 28. For most widely distributed software, however, trade secret protection alone is not sufficient for effective protection. Rival firms often allocate considerable resources to reverse engineering and decompilation, which are not violations of trade secret laws. See OTA REPORT, supra note 1, at 78-86, 146-50.

6. See, e.g., Atari Games Corp. v. Nintendo of Am. Inc., 975 F.2d 832, 838 (Fed. Cir. 1992) (finding that “[a]s literary works, copyright protection extends to computer programs and to instructions encoded on silicon chips” (citations omitted)). The federal copyright statute defines “literary works” as “works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as . . . tapes, disks, or cards, in which they are embodied.” 17 U.S.C. § 101 (1988). But see Howard G. Pollack, Note, The Gordian Algorithm: An Attempt to Untangle the International Dilemma Over the Protection of Computer Software, 22 Law & Pol’y Int’l Bus. 815, 833-34 (1991) (arguing that copyright law should be inapplicable to computer software because it is essentially utilitarian).


The most famous (or infamous) example of judicial difficulty in applying patent protection to computer software is the 1972 Supreme Court decision in *Gottschalk v. Benson.* That decision effectively precluded patent protection for computer software for many years. It questioned whether computer software was proper subject matter for patent protection and implied that such protection would require congressional action. Consequently, designers were forced to rely on copyright law to protect all aspects of software, including some potentially beyond the proper scope of copyright law. In the past few years, however, the *Benson* decision and its progeny have been reinterpreted by the courts and reexamined by academic commentators to allow patent protection for the applied ideas in computer software, as embodied in program algorithms, to become more common.

10. See id. at 67-73.
11. See Steven W. Lundberg et al., *Identifying Uncopyrightable Computer Implemented Processes and Systems,* COMPUTER LAW., April 1992, at 7; Pollack, supra note 6, at 818. However, recent decisions have begun to limit the scope of copyright protection for computer software, making stable and predictable protection for computer software even more important. See Lee T. Gesmer, *Decisions May Signify a Judicial Turnabout,* NAT'L L.J., Jan. 18, 1993, at S2.
12. See infra part III.
14. In general, an algorithm is a specific sequence of steps which can be performed by someone or something to produce a certain result. Any activity, no matter how simple or complex, can be rendered as an algorithm. As used in this Note, a program algorithm is a specific sequence of steps designed to produce a certain result when performed on a computer. See generally E. STUART LEE, ALGORITHMS AND DATA STRUCTURES IN COMPUTER ENGINEERING (1992). As discussed infra part III, a program algorithm is not a "mathematical algorithm," as defined by the Supreme Court in *Benson.*
15. See OTA REPORT, supra note 1, at 55; John T. Soma & B.F. Smith, *Software
In response to the uncertainty about the proper scope of intellectual property protection for computer software, some academic commentators have argued that the very nature of computer software makes applying either of the two "traditional" forms of intellectual property, especially patent law, conceptually or practically inappropriate. Many of these commentators have proposed that the best method of protecting computer software would be a new statutory framework, a "sui generis" law specifically tailored to the new technology. They argue, in general, that the unique dual nature of comput-


17. This is a latin phrase which means "of its own kind or class." BLACK'S LAW DICTIONARY 1434 (6th ed. 1990).

18. See Deveza, supra note 16; Dreyfuss, supra note 16; Lyons, supra note 16; Menell, supra note 16; Phillips, supra note 16; Pollack, supra note 6; Pope & Pope, supra note 16; Reichman, supra note 16; Rines et al., supra note 16; Samuelson, supra note 8; Samuelson, supra note 16; Stern, Tales From the Algorithm War, supra note 16; Stern, The Bundle of Rights, supra note 16. But see Duncan M. Davidson, Common Law, Uncommon Software, 47
er software demands such a solution. Each commentator’s vision is slightly different, but all of the proposed systems would effectively preempt traditional patent and copyright protection as applied to computer software. In addition, many of the proposals would require the creation of a governmental agency to administer the new system.¹⁹

The *sui generis* solution is premised on a key underlying assumption—conventional forms of intellectual property protection, especially patent protection, cannot be properly applied to computer software because it is unlike anything that humans have ever created. By promising to decisively resolve any conceptual and practical confusion regarding intellectual property protection for computer software, the *sui generis* proposal could have considerable appeal to Congress.²⁰

U. Pitt. L. Rev. 1037, 1080 (1986) (advocating a complicated “black box” approach involving common-law application of copyright and misappropriation principles to determine the limits of protection for computer software).

In 1984 Congress created a *sui generis* law for protection of semiconductor chip mask works. See Semiconductor Chip Protection Act of 1984, Pub. L. No. 98-620, 98 Stat. 3347 (1984) (codified as amended at 17 U.S.C. §§ 901-914 (1988 & Supp. 1992)). The Act is administered by a division of the Copyright Office and is based on a registration system, but provides for “reverse engineering” and “technological improvement” defenses in spite of proof of unauthorized copying and striking similarity. Registration under the Act creates a ten year term of protection, and provides for copyright-like remedies. There has been only one reported decision construing the Act, and its effect on the semiconductor chip industry has been minimal. See also Michael A. Ladra & James C. Otteson, *Chip-Protection Law May Miss the Mark*, Nat’l L.J., Jan. 24, 1994, at S8, S10 (noting that improved computer-assisted drafting software has made direct copying of mask works unnecessary, making the Act “irrelevant and [leaving] the protection of semiconductor chip designs ... to the patent and trade secret laws”).

The Act has been held up as a model for a *sui generis* system for intellectual property protection for computer software. See Robert W. Kastenmeier & Michael J. Remington, *The Semiconductor Chip Protection Act of 1984: A Swamp or Firm Ground?*, 70 Minn. L. Rev. 417, 465-69 & n.215 (1985) (written by the Chairman of the House Committee on the Judiciary Subcommittee on Courts, Civil Liberties and the Administration of Justice). The Kastenmeier article expresses confidence that “Congress can be trusted to consider issues arising from technological developments and to craft appropriate solutions conferring statutory protection on the creative work-product of new technologies.” *Id.* at 467; see OTA REPORT, *supra* note 1, at 27, 75-76; Samuelson, *supra* note 16, at 476. But see John A. Kidwell, *Software and Semiconductors: Why Are We Confused?*, 70 Minn. L. Rev. 533, 534 (1985) (outlining a “taxonomy of uncertainty” and concluding that computer software is far more complicated than semiconductor chips).


²⁰. The recently issued Report of the Office of Technology Assessment, a congressional agency, includes a *sui generis* proposal among the remedial options it proposes. See OTA REPORT, *supra* note 1, at 29. Two small-scale self-selecting surveys of the computer software industry’s attitude toward a *sui generis* system have been taken, with conflicting results. Compare Linda B. Samuels & Le Thi Cao, *Survey of the Opinion of Software Development Companies Concerning Intellectual Property Protection*, 32 IDEA—J.L. & Tech. 343 (1991) (indi-
This Note argues that creating a *sui generis* system is the wrong solution for several interdependent reasons. First, as explained in part II, the solution is flawed because the underlying assumption on which it is built is wrong. Although computer software is unique, each aspect of its dual nature can be understood separately in the context of prior technologies and forms of expression. Since the innovative aspect can be understood in the context of prior patentable technologies, and the expressive aspect can be understood in the context of earlier copyrightable works, the application of each kind of conventional intellectual property is entirely appropriate. Each aspect should be protected by the intellectual property system tailored and refined over many years to protect that particular aspect. To illustrate this conclusion, part II will also counter the arguments made against patent protection for computer software, which are often cited in support of a *sui generis* system.

Part III will demonstrate that the unstable legal and conceptual foundation underlying patent protection for computer software has been rectified. As a practical matter, innovative elements in computer software can be protected right now using patent law. As courts and the Patent and Trademark Office (the "PTO") have struggled to implement the *Benson* decision, they have uncovered its flaws and recognized that computer software can be patentable subject matter. In support of this conclusion, this Note analyzes the recent Court of Appeals for the Federal Circuit decision in *Arrhythmia Research Technology, Inc. v. Corazonix Corp.* The decision confirms that the deleterious effect on patent protection for computer software of the

---


unsound and ill-conceived Benson decision has now been effectively eliminated. The language of the patent statute will be the starting and ending point for deciding whether a claimed application of computer software technology is patentable subject matter.

Because the applied innovation in computer software, as embodied in the program algorithms, can now be reliably protected separately from software’s expressive aspects, the current strain on the limits of copyright protection will be greatly reduced. A stable and reliable system of protection for computer software will result from this return to the conceptual foundations of each traditional form of intellectual property, once the teachings of Arrhythmia are integrated into the PTO examination procedure. As explained in part IV, this integration will effectively address the administrative problems cited by sui generis proponents as a final reason for creating a new system. Part IV will also discuss how current problems in the patent examination system for computer software-related inventions, especially the incomplete catalog of prior art and lack of experienced examiners, can be addressed without the need for an entirely new intellectual property system.

II. CONCEPTUAL FLAWS IN THE SUI GENERIS PROPOSAL

The underlying assumption of the sui generis solution is that because computer software is unlike any technology that human society has ever known, conventional forms of intellectual property protection, especially patent law, cannot be properly applied to it. Proponents of this solution argue that the unique dual nature of the technology—as a useful art embodied in a purely expressive form—means that there are no valid parallels with the intellectual property protection applied to conventional copyrightable works or to more concrete industrial technologies.

While this argument has some appeal on its face, it misunderstands what parallels need to be drawn to decide whether conven-

25. See id. at 1057 & n.4.
26. See id. at 1061. The Arrhythmia decision was cited for just this point in Atari Games Corp. v. Nintendo of Am. Inc., 975 F.2d 832, 839 (Fed. Cir. 1992) ("In conformance with the standards of patent law, title 35 provides protection for the process or method performed by a computer in accordance with a program.").
27. See, e.g., Samuelson, supra note 8, at 1128.
28. “Art” is a term used in patent law which refers to a field of technology. See infra note 75 and accompanying text.
29. See, e.g., Samuelson, supra note 8, at 1129.
tional copyright and patent law should be applied to computer software. Because the crucial parallels can be drawn between the expressive and innovative aspects of computer software and those aspects as found in earlier human creations, copyright and patent law should be applied to computer software. When the nature of computer software is better understood, the conceptual error of the sui generis approach becomes clear.

A. The Nature of Computer Software

Although computer software is superficially embodied as lines of programming code, it is created by the programmer to directly control a computer for a useful purpose. The lines of code are only the vehicle used by the programmer to convey the binary instructions which direct the data processing operations of the hardware and actually control the computer. The actual embodiment of computer software can be thought of as an arrangement of interconnections in the computer's memory and processing units. In contrast, the conventional kinds of copyrightable expression are created merely to influence the humans that perceive them and are not useful in any other way. Because its unique dual nature gives it nearly infinite adaptability, computer software can be applied to vastly improve the performance of any human activity which can be digitally represented.

The proponents of a sui generis solution have concluded that this unique nature means that parallels with prior technologies and kinds of expression are not only conceptually inappropriate but effectively impossible. However, the essential qualities of creative expression and applied innovation found in software are the same as those found in earlier, more traditional examples of expression and innovation. Copyright and patent law have been designed and refined to protect those two different aspects of human creative effort. The fact that computer software is the first technology to embody innovative effort in an apparently written expressive form should be irrelevant to a determination of whether it is proper subject matter for patent or

30. In the form written by the programmer, the lines of code are known as “source code.” That code, when translated into machine-readable form, is referred to as “object code.” See OTA REPORT, supra note 1, at 130.
33. See, e.g., Samuelson, supra note 8, at 1128-29.
While the programming code is a kind of "useful expression," unlike traditionally copyrightable works, it is properly the subject of copyright protection because it has an expressive aspect analogous to other forms of human creative expression which is conceptually separate from the useful aspect. However, copyright law is not designed to protect the ideas underlying the expression. When those ideas are applied to perform a useful result and are sufficiently embodied and applied to define their limits, they should also be subject to patent protection.

Even though computer software is undoubtedly the beginning of a whole new field of human technology, it is still human technology. The difference in embodiment between computer software and more concrete technologies is dictated by the manner in which each technology functions. Where the mechanical engineer uses pumps and pulleys to manipulate physical objects, the software designer uses programming algorithms to manipulate inchoate data. The crucial

34. See Davis, supra note 3, at 315 (reflecting on the dual nature of computer software); Willis E. Higgins, Technological Poetry: The Interface Between Copyrights and Patents for Software, 12 Hastings Comm. & Ent. L.J. 67, 68 (1989) (same). Even opponents of patent protection for computer software concede there is no valid distinction to be made between computer software and other technologies. See, e.g., Swinson, supra note 8, at 173 ("If a process is otherwise patentable, it should be irrelevant that a computer is the intended processor.").

35. See Ralph S. Brown, Eligibility for Copyright Protection: A Search for Principled Standards, 70 Minn. L. Rev. 579, 582 (1985).

36. See Goldberg & Burleigh, supra note 16, at 297-99; see also Atari Games Corp. v. Nintendo of Am. Inc., 975 F.2d 832, 839 (Fed. Cir. 1992) (drawing an analogy between protectable expression in computer software with protectable expression found in earlier literary works).


39. See Davis, supra note 3, at 315 ("Software is a machine whose medium of con-
feature of any technology, for application of patent protection, is not how the technology works—the details of its form and manner of functioning—but the fact that it embodies innovative ideas to perform a useful purpose. Because the innovative effort in computer software is the product of the same human mental processes as the innovative efforts in more concrete technologies, valid parallels with the patent protection applied to those technologies can be drawn.

The patent system has refined definitions of novelty and nonobviousness which have been successfully applied to separate those technological advances meriting the patent prize from those that do not. These criteria are not technology-specific. Rather, they are specific to the one element common to all the arts: human innovation, embodied and applied for a useful purpose. While computer software may be difficult to understand and analyze, even by a programmer experienced in the art, it can be understood as a product of human innovation. This difficulty is not a valid reason for it not to be subject to the same system which has successfully protected and encouraged prior human technologies.

With care, the essential innovative effort invested in creating computer software can be separated from the expressive form in which the technology is embodied. As computer software evolves


One commentator has even suggested using the “functionality” doctrine found in trademark and unfair competition law. See A. Samuel Oddi, Functionality and Free Market Theory, 17 AM. INTELL. PROP. L. ASS’N Q.J. 173 (1989). Another has set out fourteen “varieties of similarity” to be used in determining whether one computer program infringes the copy-
beyond a straightforward code expression to more “organic” forms, it will be even more important that the system of intellectual property protecting computer software is based on the fundamentals of human creation—expression and innovation—rather than defined by a static sui generis system.

B. The Dynamics and State of the Software Industry Today

Proponents of a sui generis system also argue, along with certain members of the software industry, that patent protection should not be applied to computer software because the industry has flourished in the past without the benefit of strong patent protection. They point to the wide availability of inventions developed by others as one cause of this unprecedented growth. In addition, they argue that software should be treated differently than earlier technologies because the nature of computer software allows its creators to write, market and sell software in less time but with the same number of potentially patentable inventions than the creators of earlier, more concrete technologies. These commentators conclude that if all of these inventions were protected by patents, programmers would be unable to work freely.

These arguments misunderstand the purpose of the patent system, explained in part III, and implicitly fail to recognize that the software industry has matured beyond the point where developers can afford to freely share their ideas. While “borrowing” is easier than working right for another. See Davis, supra note 3, at 317-25.

42. These “organic” forms include applications such as expert systems, neural networks and artificial intelligence frameworks. See Gerald H. Robinson, Protection of Intellectual Property in Neural Networks, COMPUTER LAW., March 1990, at 17-23. But see Samuelson, supra note 8, at 1113-22 (arguing that such applications should be unpatentable because they would preempt human thought processes).

43. In contrast, semiconductor chips are a fixed and neatly defined class of creations which could be suitable for an unchanging sui generis system. See Kidwell, supra note 18, at 573.

44. See, e.g., Stallman & Garfinkle, supra note 8. It is interesting to note that this article also appears, in virtually identical form, as the “official position statement” of the League for Programming Freedom. See Against Software Patents, supra note 21, at 297.

45. See, e.g., Against Software Patents, supra note 21, at 300; Samuelson, supra note 8, at 1135-36, 1142-43.

46. See, e.g., Against Software Patents, supra note 21, at 310; Phillips, supra note 16, at 1004.

47. See, e.g., Against Software Patents, supra note 21, at 304; Samuelson, supra note 8, at 1137.

48. See, e.g., Against Software Patents, supra note 21, at 310.

around another’s patented invention, it discourages innovation by eliminating the important economic incentive of exclusivity. It also violates a fundamental premise of the American system of intellectual property, that the creator should receive the benefit of his or her creative efforts. The unprecedented growth of the software industry is a result of software’s nearly universal applicability and was achieved in spite of the lack of patent protection for the ideas underlying the software.50 Demand for any software product at all was so high in the beginning that developers could prosper without any intellectual property protection for their work and could afford to let others freely borrow their ideas. Today, the industry has matured and expanded to the point where companies need reliable protection for their efforts.51 Further serious investment will be drawn only by the security created by known and proven intellectual property rights—those found in copyright and patent law.52

*Sui generis* system proponents and their anti-patent allies have pointed to limits inherent in the platform hardware53 to argue that either copyright or patent law would stifle development of new products.54 They believe that intellectual property protection will hamper the growth of the computer software industry by blocking the unrestricted use by other programmers of the most efficient algorithm or a necessary block of code.55

These arguments are without merit. To contend that a patent on a key algorithm could block development of a profitable software product misunderstands the purpose of patent protection. It is axiomatic that an invention becomes valuable and deserving of protection precisely because it eliminates a technological limitation in an art by the novel application of the laws of physics or mathematics. Proper

13.

50. *Id.* at 17.

51. *See Within the Whirlwind, supra* note 1 (describing the growth of and fierce competition within the computer software industry).


53. The term “platform hardware” refers to the specific hardware system on which computer software is operating. This hardware is, metaphorically, the stage, or platform, from which the software performs. *See Within the Whirlwind, supra* note 1, at 5.


55. *See supra* note 54.
application of the nonobviousness criteria in § 103 of the patent statute, made while keeping in mind the limits of hardware, should ensure that only true inventions are given the protection of the patent laws. In addition, the significant transactional cost involved in obtaining a patent makes it unlikely that software developers will try to use the patent system to somehow "overprotect" their products. Finally, under the copyright law merger doctrine and other related doctrines, those sections of the software code dictated by hardware considerations can not be protected under copyright law.

Sui generis system proponents often include a mandatory licensing provision in their model statute, to "preclud[e] registration holders from withholding use of their protected expression and ideas." However, it seems unlikely that a developer who has devoted time and money to the development of a product would then turn around and refuse to either market it or license it to others. In addition, a right to a mandatory license at a fixed price, determined to be "reasonable" by a government agency, seriously impairs the bargaining position of the party subject to the involuntary license. Only if there is a flaw inherent in the fundamental structure of the relevant marketplace should the government step in to control an element as basic as price. In contrast to the railroad or utility industries, the inherent flexibility of software, the wide range of available hardware, and the number of makers of both, makes it unlikely that the computer industry is one susceptible to natural monopolies. Today, software developers employ a range of voluntary licensing options.

56. See infra notes 85-86 and accompanying text.
59. See Atari Games Corp. v. Nintendo of Am. Inc., 975 F.2d 832, 839-40 (Fed. Cir. 1992) ("If the patentable process is embodied inextricably in the line-by-line instructions of the computer program, however, then the process merges with the expression and precludes copyright protection." (citations omitted)); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983) (allowing Apple's operating system to be protected under copyright law because alternative expressions were available).
61. Id. at 1039.
62. See Within the Whirlwind, supra note 1.
63. These options include cross-licensing between firms of whole portfolios of patents.
lowing them to build on the work of others without infringing on the inventor’s or author’s rights.

Sui generis proponents argue that the computer software industry is so important to the United States economy that it deserves a separate intellectual property system specifically tailored to it. This argument misunderstands what the domestic software industry needs from United States intellectual property law: predictable, reliable, and compatible protection. Internationally, intellectual property protection for computer software has been based on copyright law. Copyright protection for foreign works is reciprocally recognized through international treaties. Recently, many countries have begun to extend patent protection to some aspects of computer software and an effort is being made to harmonize the patent systems of different countries. Compatibility with overseas intellectual property systems is crucial to the global success of the domestic software industry, an industry in which the United States leads the world, because it enables software developers to reliably protect their products overseas. In view of this need, the United States has amended its own laws several times to make them compatible with international laws.

See Cook, supra note 22, at S3.
64. See, e.g., Phillips, supra note 16, at 1002.
65. See generally Alan S. Gutterman, International Intellectual Property: A Summary of Recent Developments and Issues for the Coming Decade, 8 SANTA CLARA COMPUTER & HIGH TECH. L.J. 335 (1992) (discussing the different types of intellectual property protection available in various countries).
67. The main international copyright treaty is the Berne Convention for the Protection of Literary and Artistic Works. See OTA REPORT, supra note 1, at 104-07.
70. See OTA REPORT, supra note 1, at 94-97.
71. For example, the United States recently joined the Berne Convention. See Berne Convention Implementation Act of 1983, Pub. L. No. 100-568, 102 Stat. 2853 (1988); OTA REPORT, supra note 1, at 106-07; Carol A. Motyka, Note, Effects of U.S. Adherence to the Berne Convention, 16 RUTGERS COMPUTER & TECH. L.J. 195 (1990). In addition, Congress is considering whether to harmonize United States patent laws with those of other countries by switching to a “first-to-file” priority system, in contrast to the current “first-to-invent” system, which is virtually unique to the United States. Harmonization would also require extending the term of a utility patent from 17 years from the date of issue to 20 years from the date of filing. See H.R. 4978, 102d Cong., 2d Sess. (1992); OTA REPORT, supra note 1, at 55-56. United States commentators have been generally in favor of the proposed changes. See William S. Thompson, Reforming the Patent System for the 21st Century, 21 AM. INTELL. PROP.
Continued overseas success lies not in abandoning the balance created by the existing system, but ensuring that it is reliably and universally applied.\textsuperscript{72}

Not only is the underlying conceptual assumption of the \textit{sui generis} solution faulty, but a sound practical and theoretical legal basis for the proper protection of computer software exists today.

III. COMPUTER SOFTWARE AS PATENTABLE SUBJECT MATTER

This part will first briefly review the conceptual background of the patent laws and then examine the failure of the \textit{Benson} decision to understand that background in the context of computer software. This part concludes with a detailed analysis of the Federal Circuit's majority and concurring opinions in \textit{Arrhythmia Research Technologies, Inc. v. Corazonix Corp.}\textsuperscript{73} and the implications of those opinions.

A. The Conceptual Background

Thomas Jefferson, author of the first Patent Act, believed that "'ingenuity should receive a liberal encouragement.'"\textsuperscript{74} This belief was incorporated into the nation's fabric by its inclusion in the Constitution, which authorizes Congress to "promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."\textsuperscript{75} It stems from a policy choice which has been a cornerstone of American economic life since the nation was founded: private innovation will result in even greater rewards for the general public.\textsuperscript{76} The Patent Act of 1952 (the "Act") is the basis for the cur-

\textsuperscript{72} See Samuelson, supra note 8, at 1153 (conceding that the lack of international compatibility is a valid reason for not adopting a \textit{sui generis} system). Continued overseas success also depends on an aggressive patent strategy for United States software developers. See \textit{Americans First in Patents Since '85}, NEWSDAY, Jan. 13, 1994, at 47; Masaaki Kotabe & Eli P. Cox III, \textit{Assessment of Shifting Global Competitiveness: Patent Applications and Grants in Four Major Trading Countries}, BUS. HORIZONS, Jan.-Feb. 1993, at 57.

\textsuperscript{73} 958 F.2d 1053 (Fed. Cir. 1992).

\textsuperscript{74} See OTA REPORT, supra note 1, at 39.

\textsuperscript{75} U.S. CONST. art. I, § 8, cl. 8.

\textsuperscript{76} Since there is no record of any debate on the clause's inclusion, it seems likely that the policy choice it represents was universally held. See Higgins, supra note 58, at 316-17; Mazer v. Stein, 347 U.S. 201, 219 (1954) ("The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the tal-
rent codification of that "liberal encouragement." Applicants who meet its terms are granted, for a limited time, a right to exclude others from making, using or selling the patented subject matter. By design, the Act does not define the appropriate subject matter of a utility patent with reference to conventional technologies or traditional fields of knowledge. The universe of patentable subject matter is intended to be "anything under the sun that is made by man"—any useful application and embodiment of an idea. Since no one can predict when or in what direction science and technology will advance, the patent system is designed to encompass and protect all possibilities equally. In this way, society will get the benefit of inventions in familiar fields of knowledge as well as the benefit of those totally unforeseen—and unforeseeable—innovative leaps which create whole new fields.

79. Utility patents, which currently last for a term of 17 years, 35 U.S.C. § 154, are the type commonly used to protect the applied ideas in computer software. One other type of patent authorized under title 35, design patents, might be used to protect other aspects of computer software, such as the distinctive but ornamental aspects of the user interface. See Ken Liebman et al., The Shape of Things to Come: Design-Patent Protection for Computers, COMPUTER LAW., Nov. 1992, at 1. Design patents protect a new, original and ornamental design for an article of manufacture and last for 14 years from the date of issue. See 35 U.S.C. §§ 171-173 (1988).
80. See 35 U.S.C. § 101 ("Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."). This type of broad language has been used since 1790 to define the scope of patentable subject matter. See OTA REPORT, supra note 1, at 39. However, for administrative purposes, the PTO divides patentable subject matter into three main groups—the chemical, electrical, and mechanical arts. See PATENT AND TRADEMARK OFFICE, U.S. DEPT. OF COMMERCE, MANUAL OF CLASSIFICATION I-5 (1992 & rev. no.2 1993). Because of software's flexibility, computer software-related inventions are found in many subgroups within these main groups, especially within the electrical arts main group. Id. at I-8.
82. By allowing a technology to be subject to the patent system, Congress is necessarily implying that the existing patent system is the best one to encourage that technology to develop to its fullest potential. United States law contains only one statutory exclusion from patentability for inanimate subject matter: materials that could be used in the production of atomic weapons. See 42 U.S.C. § 2181(a) (1988). This is the only area where Congress has decided that private innovation is not in the public welfare and should not be encouraged. More controversially, Congress is considering amending title 35 to impose a two-year moratorium on the patenting of genetically engineered animals, tissue or gene sequences to allow it to consider the "ethical, legal, economic, environmental, international and social issues."
The matter sought to be patented, as described in the claims, must first meet a novelty requirement, which is designed to ensure that the invention is unknown to the public at the time the application is filed. Next, in order to ensure that society will obtain a significant benefit from each grant of exclusivity, the invention must meet a nonobviousness requirement. This requirement has been set and maintained at a fairly high level for more than 150 years. This ensures that the patent system does not protect rehashed versions of known technologies, but rather rewards truly innovative advances.

In addition, in exchange for the right to exclude others from making, using or selling the patented subject matter, the inventor must disclose in the specification the “best mode” of practicing the claimed invention known to him or her, which would allow any person skilled in the art from which the invention is drawn to make and use the invention. This requirement ensures that the public will

---


83. These are the heart of a patent, in that they "particularly point[] out and distinctly claim[] the subject matter which the applicant regards as his invention." 35 U.S.C. § 112 (1988). It is important to realize, however, that a patent does not just protect a collection of elements—it protects the idea embodied by the use of those elements. This means that an item which makes a small, inconsequential change to an element described in the claims will still probably infringe the patent. This principle is known in patent law as the "doctrine of equivalents." See Graver Tank & Mfg. Co. v. Linde Air Products Co., 339 U.S. 605, 608 (1950).


86. The wisdom of a high standard of patentability was demonstrated at the beginning of the 19th century, when for a 40-year period the United States had a ministerial registration system for patents, rather than a system in which applications are examined to see if they meet the statutory requirements before a patent is granted. For that 40-year period, a patent provided the same strong protection as a patent issued after examination, although there was no way to tell whether it was truly valid or not until it was litigated. The Senate Committee report which accompanied the Patent Act of 1836, which provided the general administrative outline of the present patent system, details the parade of frauds and frivolous litigation which resulted from the registration system. See ROBERT A. CHOATE ET AL., CASES AND MATERIALS ON PATENT LAW 76-77 (1987).

87. See 35 U.S.C. § 112. There is some controversy over the proper amount of disclo-
benefit not only by a higher level of innovation in general, but also by complete disclosure of the new invention, allowing others to use it as a basis for further research.88 The inventor must also show that the invention has been "reduced to practice," usually in the form of a working model or program.89 If an applicant fails to meet this requirement, it can be said that the invention is not sufficiently "useful" as claimed and described to merit patent protection.90

This generally applicable system has worked remarkably well in achieving its purpose, as shown by how little its fundamental outlines have been changed by Congress since 1836.91 Advances in many areas of knowledge were incorporated and protected under the patent laws because the judiciary recognized new technologies as they appeared and laid a foundation for their understanding as a "useful art" within the patent system.92 In the years prior to the Benson decision in 1972, the Court of Customs and Patent Appeals93 did this for the new technology of computer software by dismantling a judicially-created limitation on patentability known as the "mental steps" doctrine.94 It is in light of this history of success that the Benson
decision's failure of judicial understanding is particularly striking.

B. The Failure of Understanding

The reasoning in *Gottschalk v. Benson*,⁹⁵ has been described, very simply, as "monstrously bad."⁹⁶ Even one of the most ardent supporters of the result reached in *Benson* admits that the Court "did not clearly articulate the rationale for its decision."⁹⁷ The majority opinion, written by Justice Douglas, is disjointed, internally contradictory, and consists in large part of a series of unexplained quotations from prior Supreme Court opinions.⁹⁸ The best explanation for its poor quality is that the Court was somewhat ignorant and confused about the new computer software technology.⁹⁹ The Court seemed to believe that computer software functions by the direct effect of the laws of nature, coded so a computer can understand them.¹⁰⁰ In fact, computer software works by repetitively manipulating supplied data according to a series of mathematical relationships or operations defined by the human programmer—that is, by the use of algorithms.

The two claims before the Court in *Benson* encompassed an algorithm designed to improve the functioning of the computer itself.¹⁰¹ The algorithm at issue was a precisely described process for converting binary-coded decimal ("BCD") numbers to pure binary numbers,¹⁰² called the Benson-Tabbott conversion. The first claim

---

⁹⁵. 409 U.S. 63 (1972).
⁹⁶. Chisum, supra note 8, at 977-78.
⁹⁷. See Samuelson, supra note 8, at 1025 (arguing that *Benson* is still good law, despite its poor reasoning, and that there is a basis in United States patent law for denying patent protection to computer software). One industry commentator, in response to this article, drew a parallel between *Benson* and *Plessy v. Ferguson*, 163 U.S. 537 (1896). He stated that both are "fundamentally misguided opinion[s] that give[] prejudice the force of law." Heckel, supra note 49, at 16.
⁹⁸. *Benson*, 409 U.S. at 64-73.
⁹⁹. At least one commentator has also pointed to an anti-patent bias on the Court, especially on the part of the author of the *Benson* opinion. See Chisum, supra note 8, at 991 & n.118.
¹⁰¹. Id. at 65.
¹⁰². Id.
addressed by the Court covered the conversion when performed on an apparatus called a "reentrant shift register." The second claim covered the algorithm as a "data processing method," without reference to any specific computer apparatus which would carry it out. The BCD-to-binary conversion which these claims covered is a classic computer algorithm—a specific sequence of data-manipulation steps designed to be executed by a computer to accomplish a result useful only to a computer.

The Benson Court began its analysis by defining an algorithm as a "procedure for solving a given type of mathematical problem," and then found the claims at issue to encompass a "mathematical algorithm." This definition implicitly eliminated the innovative embodiment of the algorithm in the context of computer software technology. The Court did not understand the practical application of the claimed programming algorithms as computer software, and consequently it reversed the Court of Customs and Patent Appeals, and held the claims to encompass unpatentable subject matter. The confusion of the Court on this crucial point can be seen most clearly in its "nutshell" paragraph. There it calls the Benson-Tabbott conversion a "formula," as if it were simply the numerical expression of a physical law, like Einstein's general theory of relativity. By equating the two concepts—physical laws and algorithms—the Court completely discounted the painstaking human effort and the patentable innovation that went into creating the conversion algorithm. It simply did not understand that the presence of mathematical relationships and operations, in the context of computer software, should say nothing about whether an algorithm embodied in the software is patentable.

103. Id. at 73-74.
104. Id. at 74.
105. Id. at 65.
106. Id. at 73.
107. Id. at 71-72.
108. Id.
The Benson Court's ignorance regarding computer software also caused it to make several other errors. The Justices apparently believed that any claim encompassing an algorithm was necessarily overbroad—that its inventors were trying to claim more than they had invented.110 Without knowing anything about computer software design, they could not understand how specific the claims at issue actually were. The claims described an algorithm which applied mathematical relationships in a precisely defined manner to achieve a result useful in the operation of a computer. The inventors were not trying to patent the relationships in all contexts—just those described in the claims and as applied in the particular context of a computer's data conversion processes.

The discussion by the Court that the algorithm's "sweeping" end uses would "wholly pre-empt" the algorithm makes it clear that the Court did not understand that the only relevant patentability consideration was the immediate utility of the Benson-Tabbott algorithm—to improve the functioning of the computer itself.111 It was, to draw a mechanical comparison, simply a better "transmission" for connecting one part of the computer to another. The purpose to which the user put the computer, or the number of related computer models in which the invention could be used, should have been irrelevant to the patentability of the claimed algorithm. Similarly, the patentability of an improved automobile transmission would clearly not turn on the use to which the car would be put, or the number of other vehicles in which the novel apparatus could be used.

Despite the Court's express disclaimer that its purpose was not to "preclude[] a patent for any program servicing a computer," and certainly not to "freeze process patents to old technologies, leaving no
room for the revelations of the new, onrushing technology," that was the effect of the opinion. Attempts to patent computer software-related inventions dropped off dramatically. Since nearly every piece of computer software contained a "mathematical algorithm" as defined by the Court, observers logically concluded that patent protection was not available for computer software-related inventions. The Court's concluding statement, implying that the patent laws must be extended by Congress to cover computer programs, only reinforced this interpretation of the decision, especially as applied to "pure" software applications such as the one at issue.

The Benson decision began a twenty-year odyssey of judicial rationalization as later courts and the PTO tried to understand and apply the algorithm rule. The potential patentability of a computer software-related invention has turned not on whether it was sufficiently novel and nonobvious to meet the statutory standards, but on how "non-mathematical" a claimed algorithm appeared to be, and how "physical" the claimed steps were. Because these inventions were

113. See OTA REPORT, supra note 1, at 55 (number of patents granted for computer software-related inventions declined from 70 in 1972 to 5 in 1976).
114. See, e.g., Donald R. Dunner et al., Nonstatutory Subject Matter, 14 JURIMETRICS J. 112 (1973) (contemporaneous view of Benson opinion).
115. Benson, 409 U.S. at 73. The Court's apparent deference to the legislative branch is belied, however, by its quotation of three paragraphs from the 1966 Report of the President's Commission on the Patent System discussing administrative, not legal, reasons why computer software should not be subject to the patent system. Id. at 72. (citing THE PRESIDENT'S COMMISSION ON THE PATENT SYSTEM, "TO PROMOTE THE PROGRESS OF . . . USEFUL ARTS": IN AN AGE OF EXPLODING TECHNOLOGY 13 (1966), reprinted in S. Doc. No. 5, 90th Cong., 1st Sess. 21 (1967)). Its quotation of this material strongly implies that the Court was making the very policy decision it was saying only Congress could make.
so hard to protect using patent law, software developers stretched copyright law to include the "structure, sequence and organization" and "look-and-feel" of computer software.

C. The Return to the Statute

The creation of the Court of Appeals for the Federal Circuit in 1982 has brought more certainty and reliability to patent protection. Since the Supreme Court has taken no cases interpreting the patent laws for over ten years, it is reasonable to assume that the decision of the Federal Circuit will be the final judicial word on the patentability of software. The last decision of the Supreme Court concerning the patentability of a new subject matter was Diamond v. Chakrabarty. The Court held that bioengineered micro-organisms were patentable subject matter because they were the product of human ingenuity, as required by § 101 of the Patent Act.

This broad language strongly suggests that the Court has no intention of

---

Iwahashi: Part II, 74 J. PAT. & TRADEMARK OFF. SOC’Y 153, 185 (1992); Whitmeyer, supra note 13, at 1120-23.
118. See Whelan ASSOCs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1248 (3d Cir. 1986). Both the Second and Ninth Circuits have rejected the Whelan decision and held that copyright protection does not extent to the "structure, sequence and organization" of the program code. See Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 705-06 (2d Cir. 1992); Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1524-25 (9th Cir. 1992). The most effective criticism of Whelan by practitioners is found in Steven W. Lundberg et al., Baker v. Selden, Computer Programs, 17 U.S.C. Section 102(b) and Whelan Revisited, 13 HAMLINE L. REV. 221 (1990).
120. See Bender, The Renaissance, supra note 4, at 208-12; Gesmer, supra note 11, at S3.
122. 447 U.S. 303 (1980). One commentator has suggested applying the holding in Chakrabarty as a standard for computer algorithm patentability. See Kenneth C. Brooks, Comment, Human Ingenuity: A Novel Standard for Patenting Algorithms, 22 GOLDEN GATE U. L. REV. 455, 485-87 (1992). This would be redundant, however, since the ingenuity standard is drawn directly from § 101 of the patent statute, against which all claimed subject matter, including computer algorithms, should be measured.
123. Chakrabarty, 447 U.S. at 310.
involving itself in subsequent cases concerning whether or not a new technology is statutory subject matter. On the question of the patentability of computer software, the Supreme Court will undoubtedly let the determination of the experts on the Federal Circuit stand undisturbed.

The majority and concurring opinions in Arrhythmia Research Technology, Inc. v. Corazonix Corp. set out that determination. The inventors claimed, in two different ways, a process for analyzing electrocardiograph signals to determine if a high risk for ventricular fibrillation exists in patients recovering from a heart attack. This knowledge can then be used to decide whether to administer certain drugs that reduce the risk of fibrillation, but which have significant side effects. The patent had been declared invalid by the district court as directed toward nonstatutory subject matter.

The majority opinion, after setting out the invention and quoting representative claims, established its tone by quoting 35 U.S.C. § 101 and the observation in Diamond v. Chakrabarty that this statutory provision includes “anything under the sun that is made by man.” The opinion then briefly summarized Benson and Parker v. Flook, a case which extended and attempted to explain the rationale behind Benson. The court then reinterpreted Diamond v. Diehr, the last Supreme Court decision on the question of the patentability of computer software-related inventions, to have established that “when the algorithm was incorporated in a useful process, the

124. See infra note 151 and accompanying text; see also Chisum, supra note 8, at 1010-13 (arguing that Chakrabarty implied that since algorithms fall within the § 101 subject matter scope of “process,” they should be patentable subject matter absent a clear Congressional directive to the contrary).
125. 958 F.2d 1053 (Fed. Cir. 1992).
126. Id. at 1054, 1061.
127. Id. at 1054-55.
128. Id. at 1054.
129. Id. However, the Patent and Trademark Office did not even question the patentability of the claimed subject matter before granting the patent. Id. at 1055.
130. Id. at 1056 (quoting Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980)).
132. Arrhythmia, 958 F.2d at 1056.
133. 450 U.S. 175 (1981). This decision limited the reach of the Benson decision, but because the algorithm at issue was applied in connection with a conventional industrial process the effect of the holding was unclear. See also Frederick K. Longhofer, Comment, Patentability of Computer Programs, 34 BAYLOR L. REV. 125 (1982).
subject matter was statutory."134 With this holding, the opinion concluded without further discussion that "[t]he Court thus placed the patentability of computer-aided inventions in the mainstream of the law."135

The majority opinion then set out the two-step Freeman-Walter-Abele test,136 which was developed by the Court of Customs and Patent Appeals after Benson to determine whether a claimed computer software-related invention is statutory subject matter.

It is first determined whether a mathematical algorithm is recited directly or indirectly in the claim. If so, it is next determined whether the claimed invention as a whole is no more than the algorithm itself; that is, whether the claim is directed to a mathematical algorithm that is not applied to or limited by physical elements or process steps. Such claims are nonstatutory. However, when the mathematical algorithm is applied in one or more steps of an otherwise statutory process claim, or one or more elements of an otherwise statutory apparatus claim, the requirements of section 101 are met.137

The court concluded its restatement of the law by emphasizing that "what the claimed method steps do rather than how the steps are performed" is the key consideration.138 This understanding of patent law—that the useful purpose is the dispositive consideration, and not the technological expression of that purpose—is essential to establishing the patentability of algorithms within the statutory framework of title 35.

The majority opinion held that both claims were directed toward statutory subject matter.139 The process claims met the two-step test because "the steps of . . . [the] method comprise an otherwise statutory process whose mathematical procedures are applied to physical process steps."140 The apparatus claims were upheld because they were "directed to a specific apparatus of practical utility and specified application, and meet the requirements of 35 U.S.C. § 101."141 This

134. Arrhythmia, 958 F.2d at 1057.
135. Id.
136. See In re Abele, 684 F.2d 902 (C.C.P.A. 1982); In re Walter, 618 F.2d 758 (C.C.P.A. 1980); In re Freeman, 573 F.2d 1237 (C.C.P.A. 1978).
137. Arrhythmia, 958 F.2d at 1058.
138. Id. (quoting Ex parte Logan, 20 U.S.P.Q.2d (BNA) 1465, 1468 (1991)).
139. Id. at 1059-61.
140. Id. at 1059.
141. Id. at 1061.
language tied the subject matter determination as closely as possible, in the context of the Benson rule, to unqualified statutory subject matter.\textsuperscript{142} Since any claim covering the applied ideas—the programming algorithms—used in computer software is now likely to meet the two-step test as applied by the majority opinion, it is now undeniably true that "the patentability of computer-aided inventions [is] in the mainstream of the law."\textsuperscript{143}

The minority opinion, although it acutely limited the Benson rule, did not directly challenge the conceptual foundation for it. Judge Rader's opinion in Arrhythmia, concurring in the result, did just that by finding that the Supreme Court in Diehr did not simply limit but implicitly rejected the Benson rule.\textsuperscript{144} He read the opinion in Diehr to have "cut the Gordian knot" of precedent "encircling and confining the Benson rule" by relying for its result primarily on the language of the patent statute, rather than the contradictory Freeman-Walter-Abele test.\textsuperscript{145}

The concurring opinion began by chronicling the struggle of subsequent courts to implement the Benson rule.\textsuperscript{146} Judge Rader explained the struggle by illustrating the confusion caused by the contradictory definitions of "the terms invoked to preclude patentability."\textsuperscript{147} He then quoted Justice Frankfurter's helpful discussion of this very issue: "Arguments drawn from such terms [as "the work of nature" or the "laws of nature"] for ascertaining patentability could fairly be employed to challenge almost every patent."\textsuperscript{148} The concurrence concluded that "[w]hen attempting to enforce a legal standard embodied in broad, vague, nonstatutory terms, the courts have floundered."\textsuperscript{149}

He also criticized the Freeman-Walter-Abele test, designed to

\textsuperscript{142} The majority opinion has also been interpreted as refining and streamlining, but not effectively discarding, the two-step test for subject matter determinations in the context of computer software-related algorithms. See Alan D. Minsk, The Patentability of Algorithms: An Update on the Status of the Current Doctrine, 9 SANTA CLARA COMPUTER & HIGH TECH. L.J. 233, 233-38, 246 (1993); David S. Benyacar, Mathematical Algorithm Patentability: Understanding the Confusion, 19 RUTGERS COMPUTER & TECH. L.J. 129, 174-79 (1993).

\textsuperscript{143} Arrhythmia, 958 F.2d at 1057.

\textsuperscript{144} Id. at 1061, 1064-65.

\textsuperscript{145} Id. at 1061.

\textsuperscript{146} Id. at 1062 n.1.

\textsuperscript{147} Id. at 1062-63 & n.2 (discussing the contrasting meanings of "formulae," "law of nature," "natural phenomena," and "algorithm").

\textsuperscript{148} Id. at 1063 (quoting Funk Bros. Seed Co. v. Kalo Inoculant Co., 333 U.S. 127, 134-35 (1948) (Frankfurter, J., concurring)).

\textsuperscript{149} Id. at 1063.
implement the Benson rule, as inherently vague and unpredictable. Judge Rader advocated reading Diehr as instructing lower courts to give the categories used in the Patent Act their "literal and predictable meaning, [as was done in Chakrabarty,] without conjecturing about the policy implications of that literal reading." He found the claimed subject matter patentable simply because the claims described a "practical and potentially life-saving process . . . .[r]egardless of whether [it is] performed by a computer." The combined weight of the two opinions in Arrhythmia makes it clear that computer software, like any useful technology, is now subject to the full protection and encouragement of the patent system. The concurring opinion went so far as to advocate expressly discarding the misconceived Benson rule and the tortured two-step test and returning to the language of the statute as the first, last and only determinant of whether computer algorithms are patentable subject matter. By restoring patent law to its proper place, copyright can be returned to protecting only the truly expressive aspects of computer software.

With the sound conceptual and legal basis for intellectual property protection of computer software in place, the two Offices charged with administration of the patent and copyright laws will be able to do their jobs efficiently and effectively.

IV. THE ADMINISTRATION OF PROTECTION

Some of the reasons advocated for a sui generis system and against the current system are based on problems which have developed regarding the administration of the current intellectual property

150. Id. ("Thus, the court apparently made compliance with the two-part test a function of the 'significance' of additions to the algorithm—hardly a predictable standard.").

151. Id. at 1064.

152. Id. at 1066.

153. See Atari Games Corp. v. Nintendo of Am. Inc., 975 F.2d 832, 839 (Fed. Cir. 1992) (Rader, J.) ("In conformance with the standards of patent law, title 35 provides protection for the process or method performed by a computer in accordance with a program.").


155. See Atari Games Corp., 975 F.2d at 839 ("Thus, patent and copyright laws protect distinct aspects of a computer program."); Brown, supra note 35, at 580 (emphasizing that "the integrity of future copyright law depends on insuring that changes in the law do not occur at the expense of distorting the policies underlying copyright").
system for computer software, and specifically in the examination of patent applications claiming computer software-related inventions by the PTO.\textsuperscript{156} Basically, in the examination of a patent application, the Patent and Trademark Office examiners compare the claimed invention with each of the criteria codified in the Patent Act.\textsuperscript{157} To do this, they use the current state of judicial understanding regarding those criteria and the PTO's internal catalog of prior art,\textsuperscript{158} as supplemented by the applicant.\textsuperscript{159}

There have been three main areas of difficulty in the examination process for computer software-related inventions. While the cause of many of these problems can be traced to the Benson algorithm rule,\textsuperscript{160} even advocates of patent protection for computer software agree they need to be addressed.\textsuperscript{161}

First, the staff of examiners in this field of technology has been inadequate to effectively evaluate applications claiming computer

\textsuperscript{156} See Against Software Patents, supra note 21, at 303, 305-07.

The Copyright Office has also had a few difficulties, although the more ministerial nature of the job that Office performs limits the number and magnitude of the potential problems. The Copyright Office oversees a registration system, maintaining a file of copyrighted expression for the eventualty of litigation. It does not need to compare the submissions with existing copyrighted works or make complicated technical determinations. See 17 U.S.C. § 102 (1988 & Supp. 1992) (stating that "copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression"); 17 U.S.C. §§ 409-410 (1988 & Supp. 1992) (listing requirements for a certificate of registration). Consequently, its smooth and effective functioning does not depend on the completeness of its catalog of previously copyrighted expression or any advanced technical expertise on the part of its officials.

What problems the Copyright Office has suffered arise from the sheer volume of applications and the nature of the expression itself—it is not possible to determine whether a listing of programming code contains "original authorship" merely by visual inspection. Consequently, the Copyright Office issues a kind of conditional registration to computer software works. Under what it calls the "rule of doubt" procedure, the burden is shifted to the courts to make the "original authorship" determination should the copyright be litigated. See OTA REPORT, supra note 1, at 66; 37 C.F.R. § 202.20(c)(2)(vii)(B) (1993).

\textsuperscript{157} See 37 C.F.R. §§ 1.11-1.825 (1993) (setting out procedures to be followed by the PTO); PATENT AND TRADEMARK OFFICE, U.S. DEPT. OF COMMERCE, MANUAL OF PATENT EXAMINING PROCEDURE (5th ed. 1993) (same).

\textsuperscript{158} The term "prior art" includes "any relevant knowledge, acts, descriptions and patents which pertain to, but predate, invention in question." BLACK'S LAW DICTIONARY 1193 (6th ed. 1990).

\textsuperscript{159} The applicant can bring previously uncatalogued references to the attention of the examiner to explain why a patent meets the statutory criteria. See 37 C.F.R. §§ 1.97-1.98 (1993).


\textsuperscript{161} See, e.g., Heckel, supra note 49, at 16.
software-related inventions. To do their job, the PTO examiners must be experts in the art from which a claimed invention is drawn in order to understand the invention and apply the statutory criteria to it. The initial staff of examiners charged with examining applications claiming software-related inventions was composed almost entirely of electrical engineers, and quite understandably seized on the Benson algorithm rule to avoid making determinations in an area in which they had no experience. Since the signals that the PTO had received from both Congress and the Supreme Court indicated that copyright law was the only form of intellectual property protection realistically available for computer software, no serious effort had been made, until recently, to recruit and retain truly qualified examiners in the art of computer software. As a result, a generally unreceptive attitude toward these applications was fostered.

Second, because the Benson algorithm rule made computer software-related inventions so difficult to patent, a comprehensive catalog of prior art does not exist, making determinations of novelty and nonobviousness very difficult. Since the Benson rule was created just as the software industry began to take off and the greatest number of foundational inventions were being made, the catalog could not begin in the usual fashion with the patents on those breakthrough inventions. Without a complete catalog, one that reflects the current state of the art, the examiners cannot easily determine whether a claimed invention would be obvious to a practitioner of average skill in the relevant art, or even whether the claimed invention is novel at all. The combined effect of these two handicaps had made the PTO even more reluctant to make difficult decisions about the patentability of software.

162. Id. In addition, until recently the PTO did not allow people with degrees in computer science or mathematics to sit for the patent bar. See Samuelson, supra note 8, at 1138 n.442.


164. See supra part III.B.

165. See OTA REPORT, supra note 1, at 55-56.

166. See id. at 6-7. This handicap has been compounded by the computer software industry’s use of trade secret protection for some programs, and the highly informal manner that knowledge concerning advances in the industry was initially disseminated among programmers. Id. at 8 (Box 1-B).

167. These are the statutory requirements for patentability set out in sections 102 and 103 of title 35. See supra notes 84-86 and accompanying text.
Third, the Benson algorithm exclusion and the decisions attempting to apply it have created a labyrinthine rule, the Freeman-Walter-Abele test, which, not surprisingly, the PTO found difficult to apply consistently.168 Consequently, a significant backlog of applications has built up, creating the danger of "landmine" patents issuing several years after the applications were filed.169 The rapid pace of development in the industry means that a promising software project could suddenly become an infringing one, adding uncertainty and discouraging investment in novel types of software.170 In addition, the examination of applications in this area has focused on whether the claimed invention is patentable subject matter, not on whether it meets the novelty and nonobviousness criteria, which are usually the crucial determinations to be made in evaluating patentability. This has cast doubt on the validity of some already-issued patents.171

The proponents of a sui generis system claim that their proposed solution will eliminate all of these handicaps.172 This is difficult to rebut, as the administrative structures of proposed sui generis systems are always described in ideal terms.173 As an initial matter, it would take a considerable amount of time and effort to get a new administrative system up and running. During that time, uncertain investors might not invest in the software industry, as they waited to see how the courts and the new administrative agency would interpret and apply the new statutory framework.174 In addition, any sui generis system specifically tailored to the state of the art when the law was enacted would undoubtedly quickly become outdated as computer software's rapid technological progress outpaced the fixed provisions of the new law.

168. See supra notes 116, 117, 146-149 and accompanying text.
169. See OTA REPORT, supra note 1, at 9 (Box 1-C).
170. See id.; Samuelson, supra note 8, at 1136.
171. An excellent example of this is the potentially very broad patent issued to a small California company, Compton's New Media, covering a multimedia search-and-retrieval system. See Cook, supra note 22, at S2. In an unusual move, the Commissioner of the PTO has ordered a reexamination on his own initiative of the validity of the patent. Id. at S4 n.1; see also 35 U.S.C. § 303(a) (1988); infra note 180 and accompanying text. If the patent is upheld, it could become one of the breakthrough patents in the new field of interactive multimedia technology.
173. See, e.g., Rines et al., supra note 16, at 6-7 (first stating that "[t]he INP Act will have an easy and inexpensive registration system" but then going on to list six different items that must be submitted to the new agency before the registration will issue, including "a statement describing in detail the developer's contribution to the art").
174. See OTA REPORT, supra note 1, at 31.
Moreover, assuming that the proposed *sui generis* system involves an examination procedure, there is no reason to believe that it would function any more efficiently than the existing PTO will once the problems detailed above have been completely resolved.\textsuperscript{175} There are a number of indications, discussed below, that the PTO will work effectively to apply the patent system to computer software-related inventions in the future.

The conceptual clarification and virtual elimination of the *Benson* rule accomplished by the *Arrhythmia* decision will do a tremendous amount to alleviate the handicaps the PTO currently bears in examination of applications in this area. Since extended evaluation of the subject matter with regard to the two step *Freeman-Walter-Abele* test is no longer required, examiners will be able to focus on the statutory criteria for patentability—novelty and nonobviousness—which the PTO has a great deal of experience in applying in the context of other technologies. This should not only improve the quality of the new prior art being added to the catalog but also reduce the amount of time required between the application for and issuance of a patent. The positive signal sent by the Federal Circuit in the *Arrhythmia* decision should make the PTO more receptive to applications covering computer software-related inventions, and eliminate the need to resort to appellate review to surmount a subject matter rejection.\textsuperscript{176}

In addition, a number of other options are available to rectify existing problems without resorting to a radical *sui generis* solution. The PTO’s catalog of prior art can be supplemented by earlier publications and descriptions of various software applications.\textsuperscript{177} Congress can provide funding commensurate with the importance of computer software to the United States economy, allowing the PTO to hire and

\textsuperscript{175} Conversely, if the proposed system is primarily a registration one, there is no reason to think that a new agency would be any more efficient than the Copyright Office is right now, especially in light of the experience that Office has both with computer software and registration systems in general.

\textsuperscript{176} *But* see D.C. Toedt III, *Software Patent Controversies Lead to Different Outcomes in the Federal Circuit, PTO*, COMPUTER LAW., July 1992, at 18, 18-19. The only way to ensure a receptive approach to applications for patents covering computer software-related inventions is the appointment of a sympathetic Commissioner. See Victoria Slind-Flor, *New Patent Chief Reinvents His Job*, NAT’L LJ., Feb. 28, 1994, at 1, 40 (discussing the untraditionally active role Bruce A. Lehman, the recently appointed Commissioner, has taken to explore the effect patents may have on the computer software industry).

\textsuperscript{177} A group of software industry participants has created an organization called the Software Patent Institute to address this problem. *See OTA REPORT*, *supra* note 1, at 56; *Cook, supra* note 22, at 33 (listing Apple Computer, Microsoft, IBM, and Lotus Development Corp. among the contributing companies).
retain more and better qualified examiners.\textsuperscript{178} Use of the recently created reexamination procedure will allow the computer software industry to quickly establish the validity of issued patents without plenary litigation.\textsuperscript{179} If litigation is required to contest the validity of already issued patents, the use of special masters and magistrates with greater technical expertise to try patent cases could make resolution of infringement questions faster and more predictable.\textsuperscript{180}

V. CONCLUSION

As demonstrated above, computer software technology fits conceptually within the context of prior human creative works and technologies. The copyright and patent systems have worked effectively to protect and encourage earlier, more conventional human efforts, and there is no valid conceptual or practical reason to think these systems will not do the same for computer software. The law, as evidenced by the \textit{Arrhythmia} decision, has established a stable foundation for the new technology and can accommodate it without further delay. At this stage, imposition of an untested and potentially inflexible \textit{sui generis} system would be unnecessary and even dangerous to the continued success of the computer software industry.

\textit{John M. Griem, Jr.}

\begin{footnotesize}
\textsuperscript{178} See Goodman, supra note 52, at 178-79.

\textsuperscript{179} See 35 U.S.C. §§ 301-307 (1988) (allowing anyone to ask the PTO to reexamine a currently enforceable patent); Higgins, supra note 58, at 318-19.

\textsuperscript{180} A more controversial but effective way to make patent litigation more efficient and predictable would be to eliminate the option of trial by jury for patent infringement cases. There is disagreement over whether the Seventh Amendment requires that patent infringement suits, fundamentally equitable actions, be tried by a jury. Many practitioners feel that the complex technical determinations that must be made in a patent case are ill-suited to determination by an untrained jury. Others note that this could be unfair to small independent inventors because jurors tend to favor them when confused by the technical issues and the law. The Court of Appeals for the Federal Circuit has accepted three cases for review to examine this issue. See Richard B. Schmitt, \textit{Juries' Role in Patent Cases Reconsidered}, \textit{Wall St. J.}, Feb. 18, 1994, at B6, B6.
\end{footnotesize}