

Fall 1997

Legal Aspects - Software Reverse Engineering and Copyright: Past, Present and Future, 31 J. Marshall L. Rev. 1 (1997)

Brian C. Behrens

Reuven R. Levary

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



Part of the [Computer Law Commons](#), [Intellectual Property Law Commons](#), [Legislation Commons](#), and the [Science and Technology Law Commons](#)

Recommended Citation

Brian C. Behrens & Reuven R. Levary, Legal Aspects - Software Reverse Engineering and Copyright: Past, Present and Future, 31 J. Marshall L. Rev. 1 (1997)

<https://repository.law.uic.edu/lawreview/vol31/iss1/1>

This Article is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in UIC Law Review by an authorized administrator of UIC Law Open Access Repository. For more information, please contact repository@jmls.edu.

ARTICLES

LEGAL ASPECTS - SOFTWARE REVERSE ENGINEERING AND COPYRIGHT: PAST, PRESENT AND FUTURE

BRIAN C. BEHRENS* & REUVEN R. LEVARY**

With technological advances in the field of computers, the legal world is discovering that the laws that apply to this field must advance as well. Copyright laws that currently govern the development of computer software are still quite ambiguous, with difficulties in their application and debates over the scope of the law. In light of conflicting case law, it is troublesome for an attorney to give clients advice on whether certain software development techniques violate copyright laws or carry other legal consequences.

What is certain is that the courts are deciding more and more computer software cases.¹ As a result, future law regarding copyright infringement in this area is still subject to change. However, this does not make understanding how copyright laws apply to software any easier today. Courts continue to have difficulty understanding the computer technology, which is often the subject of this type of litigation, and judges are unfamiliar with the methods

* Brian C. Behrens is an attorney with Suelthaus & Walsh, P.C., in St. Louis, Missouri, practicing in the Corporate, Business, Securities and Intellectual Property Law departments. He earned both his J.D. (*cum laude*) and his M.B.A. from Saint Louis University, after graduating from the University of Missouri - Columbia with a bachelor's degree in business administration (B.S.B.A. *cum laude*).

** Reuven R. Levary is Professor of Decision Sciences at Saint Louis University. He received a Ph.D. in operations research from Case Western Reserve University. Dr. Levary is co-author of *Quantitative Methods for Capital Budgeting*, South-Western Publishing Co., 1990, and editor of *Engineering Design: Better Results Through Operations Research Methods*, North-Holland Publishing Co., 1988. He has published widely in a variety of professional journals. Dr. Levary serves on the editorial boards of several journals. He is a Fellow of AAAS and a member of INFORMS, IEEE, and the Society for Computer Simulation.

1. Note that throughout this article the terms software and computer program are used interchangeably. These terms both refer to the written language that instructs a computer as to what functions and actions to run.

and procedures that computer companies use to develop software. As more cases are decided, courts are becoming more familiar with the concepts and issues being litigated and more aware of the impact judicial decisions have over the entire computer industry.

This article addresses various issues surrounding computer software development and copyright law. It examines the legal aspects of software copyright law, exploring its past, present, and future, while paying particular attention to a relatively recent phenomenon known as reverse engineering. Part I provides a basic overview of copyright law, including the history and policy considerations behind copyright law. Part II examines the methods computer programmers use to develop new software that interacts with existing programs, enhances existing programs or directly competes with existing programs. Part III details two important cases involving reverse engineering: *Sega Enterprises Ltd. v. Accolade, Inc.*² and *Atari Games Corp. v. Nintendo of America, Inc.*³ Part IV discusses the fair use statutory defense to a claim of copyright infringement and advocates its continued application in software development cases as a means of promoting progress in the computer industry and reducing the amount of litigation over copyright infringement. Finally, this article argues that copyright law, as it relates to computer software, should be amended to define the rights of copyright holders with greater specificity and in a manner that is more useful to practitioners. Computer software is a type of creative work that differs greatly from traditional literary works. As a result, this article also asserts that a new system of laws needs to be developed to cater specifically to the computer software industry.

I. COPYRIGHT LAW

Ever since Congress drafted the original Copyright Act in 1909,⁴ traditional copyright law has focused on protecting literary works.⁵ With the advent of computers and computer software, disputes arose over how to protect this new technology. The Copyright Office decided that computer programs were similar in nature to books and began protecting these programs as literary works in 1964.⁶ Some years later, major revisions in copyright law

2. 977 F.2d 1510 (9th Cir. 1992).

3. 975 F.2d 832 (Fed. Cir. 1992).

4. Copyright Act of 1909, ch. 320, 35 STAT. 1075 (1909).

5. The Constitution grants Congress the power to enact laws "[t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." U.S. CONST. art. I, § 8, cl. 8.

6. See Evan Finkel, *Copyright Protection for Computer Software in the Nineties*, 7 SANTA CLARA COMPUTER & HIGH TECH. L.J. 202 (1991).

took place, resulting in the Copyright Act of 1976.⁷ Shortly thereafter, a congressionally sponsored study by the National Commission on New Technological Uses of Copyrighted Works (CONTU) was finalized in a 1978 report.⁸ This report ultimately concluded that copyright law was indeed the best way to protect intellectual property rights in computer software.⁹ In 1980, Congress incorporated CONTU's suggestions into the newly amended 1976 Act.¹⁰ Today, computer software is protected as a category of literary works.

Unfortunately, CONTU's report did not make analyzing software copyright infringement claims any easier. Software developers, attorneys and judges alike have had a difficult time applying copyright law to computer software. The difficulty arises because of the nature of computer software. Computer programs are similar to books and other literary works in that they are written, creative products of an intellectual nature. Additionally, computer software, like other written works, are easily duplicated and reproduced.¹¹

However, computer programs are substantially different from traditional literary works in that words in computer programs are not in human readable form. Words in computer programs are written in cryptic languages that only a computer can decipher. Further, computer programs are more than just words, they are instructions which enable a computer to carry out specific functions and complex tasks.¹² Programmers write software in high-level languages referred to as source code.¹³ Source code closely re-

7. See generally 17 U.S.C. §§ 101-914 (1994).

8. In order to explore the intricacies of copyright protection for computer software, Congress formed the National Commission on New Technological Uses of Copyrighted Works in 1974. Finkel, *supra* note 6, at 203.

9. NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORDS, FINAL REPORT 13 (1979).

10. The copyright statute broadly defines literary works to include works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks or cards, in which they are embodied. 17 U.S.C. § 101 (1994).

11. Gary R. Ignatin, Comment, *Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility*, 140 U. PA. L. REV. 1999, 2004 (1992). Ignatin notes CONTU's suggestion that computer software should be governed by copyright law primarily because software can be easily copied with little expense. Raymond T. Nimmer & Patricia A. Krauthaus, *Copyright and Software Technology Infringement: Defining Third Party Development Rights*, 62 IND. L.J. 13, 21 (1987).

12. Copyright law defines 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. 17 U.S.C. § 101 (1994).

13. Examples of these high-level languages, or machine languages include BASIC, FORTRAN, COBOL, Ada, C, and Pascal to name a few. Ignatin, *supra* note 11, at 2001-02 & n.8. Source code is a programming language com-

sembles the English language. After the source code is completed, the programmer uses a compiler to convert the program into object code, a machine language consisting of all ones and zeroes.¹⁴ Once a program is converted into object code, one cannot determine how a program functions without taking steps to reverse this non-readable language back into source code.

This conversion process presents many legal questions. First, copyright law only protects an author's original expression of a given idea and does not protect "any idea, procedure, process, system, method of operation, concept, principle or discovery, regardless of the form in which it is described . . . [or] illustrated."¹⁵ An author must obtain a patent to protect an idea from unauthorized use. Increasingly, the argument is being made that computer programs are not simply the expression of an idea, but a collection of functions and processes to which copyright law provides little protection from copying. Herein lies the great debate: how to rationally separate the expression of ideas in a computer program from the functions that a computer program carries out. Courts have been tangling with the idea-expression dichotomy since the case of *Baker v. Selden* in 1879.¹⁶ Unfortunately, there appears to be no easy answer to this question. Judge Learned Hand noted that "no principle can be stated as to when an imitator has gone beyond copying the idea, and has borrowed its expression . . . [d]ecisions must therefore inevitably be ad hoc."¹⁷

Second, even if an idea or function carried out in a computer program is not protected by copyright law, a question still remains. Is it a violation of copyright law to copy a computer program and translate it from object code back to source code? This question will arise with greater frequency as we move into an age where many everyday products incorporate computer chips in their design.¹⁸ Courts have to devise some type of framework in which to decide these cases until Congress drafts a new statutory scheme, such as a *sui generis* system often proposed by commentators.¹⁹

posed of combinations of words and mathematical equations. Marshall Leaf-fer, *Engineering Competitive Policy and Copyright Misuse*, 19 U. DAYTON L. REV. 1087, 1090 (1994).

14. See BRYAN PFAFFENBERGER, *QUE'S COMPUTER USER'S DICTIONARY* 424 (3d ed. 1992).

15. 17 U.S.C. § 102(b) (1994).

16. 101 U.S. 99 (1879).

17. *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960).

18. For example, most of today's newer automobiles have some type of computer chip, that controls the automobile's engine performance. The chips stores other diagnostic codes in memory to better enable mechanics to assess any problems in need of maintenance.

19. The rights protected under a *sui generis* utility model "would be more like those of patent law than copyright law." Richard H. Stern, *A Sui Generis Utility Model Law As An Alternative Legal Model for Protecting Software*, 1 U.

The recent statutory changes in copyright law may not be enough to accommodate the rapidly changing world of computer technology. With these issues presented as a backdrop, reverse engineering will now be examined in greater detail.

II. REVERSE ENGINEERING

A significant amount of litigation involving computer software focuses around the issue of reverse engineering. Nonetheless, it remains one of the most misunderstood concepts by the courts, legal counsel and executives in the computer industry.²⁰ The United States Supreme Court defined reverse engineering as "a fair and honest means . . . [of] starting with the known product and working backwards to divine the process which aided in its development or manufacture."²¹ In 1989, a United States District Court defined reverse engineering as "the process of starting with a finished product and working backward to analyze how the product operates or how it was made."²² Essentially, reverse engineering is the method by which computer programmers study an existing program in machine-readable code by breaking it down into human readable form to create a similar product, or one that can be used in conjunction with the existing software. This process of converting a program from object code back into source code is known as decompilation or disassembly.

Some computer software companies claim reverse engineering is unlawful because it is far too easy to take an existing program, use reverse engineering to discover how it operates, make slight modifications to the original program and market this amended version as a new software product.²³ Indeed, if a second program is developed which is substantially similar to the original, with more in common than just the functional operations, it is most likely a case of software copyright infringement.²⁴

BALT. INTELL. PROP. L.J. 108, 113 (1993). "It typically has a narrower scope, it may be allowable upon mere registration, its validity is sustained on the basis of a lesser showing of technical merit than a regular patent, and it may have a shorter life span." *Id.* at 112.

20. G. Gervaise Davis, III, *Scope of Protection of Computer-Based Works: Reverse Engineering, Clean Room and Decompilation*, in 15TH ANNUAL COMPUTER L. INST., at 115 (Patent, Copyrights, Trademarks, and Literary Property Course Handbook Series, 370 PLI/PAT 115, Oct./Nov. 1993).

21. *Kewanee Oil Corp. v. Bicron Corp.*, 416 U.S. 470, 476 (1974).

22. *Secure Serv. Tech., Inc. v. Time & Space Processing*, 722 F. Supp. 1354, 1361 n.16 (E.D. Va. 1989). Unlike *Kewanee*, which dealt with trade secrecy, *Secure Services Technology* focused directly on computer software copyright infringement.

23. David P. Hamilton, *U.S. Criticizes Japan on Panel Software*, WALL ST. J., Nov. 10, 1993, at B5.

24. Copyright law protects the expression of an author's work, not the underlying functions of a program that can only be protected by patent law. Apparently some courts confuse the expression-function classification by finding

More recent cases however, discussed in Part III, should curtail this fear as certain requirements must now be met before an altered computer program can be marketed as a new product.²⁵ The truth is that not all reverse engineering efforts are illegal. In fact, most of the time they are not. There are numerous reasons why programmers use reverse engineering and producing a competitive program is only one of them. Reverse engineering is used to teach students how to write programs,²⁶ to repair malfunctioning software,²⁷ to produce similar software that runs on different computer systems,²⁸ to modify a program for use on one's own computer²⁹ and to develop software that operates in conjunction with original software.³⁰

III. RECENT CASE LAW

The goal of copyright law is two-fold. First, copyright law is intended to promote the freedom of information and advancement of ideas that may benefit society. Second, copyright law aims to protect an author's interest in his creative work by giving him control over its dissemination and allowing him to reduce the exploitation of his work. Although these interests must be balanced, the Supreme Court has clearly noted that "[t]he primary objective of copyright is not to reward the labor of authors, but [t]o promote

similarities on the basis that two separate programs perform the same functions. See generally Davis, *supra* note 20 (citing Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2nd Cir. 1982) and Gates Rubber Corp. v. Bando Am., Inc., 798 F. Supp. 1499 (D.C. Colo. 1992) as cases which note this problem).

25. *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992); *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832 (Fed. Cir. 1992). See also 17 U.S.C. § 107 (1994) (setting forth requirements which must be complied with before the fair use defense to copyright infringement can be invoked).

26. Davis, *supra* note 20, at 142-45. Davis sets forth non-commercial and commercial reasons for reverse engineering. *Id.* Davis notes that use of such copyrighted materials for educational purposes is permitted by 17 U.S.C. § 107. *Id.*

27. Davis, *supra* note 20, at 142-43. Davis contends this use would appear to be lawful under 17 U.S.C. § 117. *Id.*

28. For example, in order to adapt a word processor that runs on a Macintosh computer to an IBM Windows environment, one must be able to determine how the Windows system functions.

29. *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 270 (5th Cir. 1988) (holding such modification of software to be legal when the original program was purchased and owned lawfully). See also *Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc.*, 780 F. Supp. 1283, 1298 (N.D. Cal. 1991), *aff'd*, 964 F.2d 965 (9th Cir. 1992) (reasoning that modification of a software program for use on one's own computer is legal where software is purchased and owned lawfully).

30. See *Lewis Galoob Toys*, 780 F. Supp. at 1287-91 (holding that reverse engineering was appropriate to aid in design of Game Genie, a video game cartridge attachment that temporarily altered certain aspects of the Nintendo game).

the Progress of Science and useful Arts."³¹

Recently, the courts have indeed shifted their emphasis from the secondary goal to the primary goal - providing information to the public to advance science and the useful arts by restricting the scope of what copyright law protects. Two cases in particular, *Sega Enterprises Ltd. v. Accolade, Inc.*³² and *Atari Games Corp v. Nintendo of America, Inc.*,³³ are direct examples of this changing perception and provide some instruction to lower courts deciding similar copyright infringement cases. In addition, these decisions will serve to better guide attorneys on how to advise corporate clients who consider using reverse engineering.

A. Sega Enterprises Ltd. v. Accolade, Inc.

1. Procedural History

Sega Enterprises Ltd. is a producer and marketer of home video game systems and game cartridges.³⁴ Accolade does not manufacture its own video game systems, but instead develops video games for use with video game systems of other companies.³⁵ Accolade wished to develop game cartridges for use in the Sega Genesis video game system. However, Accolade did not obtain a license from Sega to develop the game cartridges.³⁶ In order to produce games that were compatible with the Genesis system, Accolade purchased several Sega game cartridges and reverse engineered them to reveal the games' source codes and to discern how the games functioned.³⁷ After uncovering the Sega cartridges' source code, Accolade developed its own software for use on the Genesis system.³⁸ In 1990, Accolade began selling its own cartridges.³⁹

As other companies began to produce games that competed with Sega games, Sega developed another version of its video game system titled Genesis III.⁴⁰ This new game system included a feature known as the Trademark Security System (TMSS) that prevented any competitor's game cartridge from working on the new

31. *Feist Publications, Inc. v. Rural Tel. Serv.*, 499 U.S. 340, 349 (1991) (quoting the language of U.S. CONST. art. I, § 8, cl. 8).

32. 977 F.2d 1510 (9th Cir. 1992).

33. 975 F.2d 832 (Fed. Cir. 1992).

34. *Sega*, 977 F.2d at 1514.

35. *Id.*

36. *Id.* Although Sega had granted other video game companies a license to produce and market games for use with the Genesis system, Accolade did not try to obtain a license so as to avoid paying royalties to Sega. *Id.*

37. *Id.*

38. *Id.* at 1515.

39. *Sega Enters. Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392, 1395 (N.D. Cal. 1992).

40. *Sega*, 977 F.2d at 1515.

system unless the cartridge contained a certain code.⁴¹ Sega gave this code only to those companies it granted licensure.⁴² Upon discovering that its games did not run on the Genesis III system, Accolade purchased Genesis III game cartridges that were compatible with the system, reverse engineered them to obtain the new code and implemented this code into its own games.⁴³

After discovering that Accolade copied the code for use in its games, Sega filed a copyright infringement suit against Accolade.⁴⁴ Sega sought to enjoin Accolade from reverse engineering any more of its games or manufacturing any other games containing the reverse engineered code.⁴⁵ The district court granted an injunction in favor of Sega reasoning that Accolade violated Sega's exclusive right to reproduce the object code by making copies of the original games for purposes of reverse engineering.⁴⁶

The Ninth Circuit Court of Appeals granted Accolade's request for an interlocutory appeal and dismissed the lower court's preliminary injunction.⁴⁷ On appeal, Accolade set forth four arguments: 1) intermediate copying is not a copyright infringement if the end product is not substantially similar to the copyrighted program;⁴⁸ 2) disassembly of a computer program is not an infringement of copyright law when it is used to access the program's ideas and functions pursuant to 17 U.S.C. § 102(b);⁴⁹ 3) copying a program is authorized by 17 U.S.C. § 117, if it is an essential step in the utilization of the computer program in connection with a machine;⁵⁰ and, 4) disassembly of a computer program is lawful under the fair use exception of 17 U.S.C. § 107.⁵¹ Ultimately, the Ninth Circuit was not persuaded by the first three arguments, but agreed that disassembly was protected under the fair use exception to copyright law.⁵²

41. *Sega*, 785 F. Supp. at 1395.

42. *Id.*

43. *Id.* When a cartridge was placed in the Genesis III system, the system would search for the TMSS code in the cartridge, and if found, the game monitor would display the following message: PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD. *Sega*, 977 F.2d at 1515. If the proper code was not found, the game would not run. *Id.* Because the message incorrectly stated that Accolade had licensure from Sega, Sega later sued Accolade for trademark infringement. *Id.* at 1516.

44. *Sega*, 977 F.2d at 1516.

45. *Id.*

46. *Sega*, 785 F. Supp. at 1396.

47. *Sega*, 977 F.2d at 1517.

48. *Id.*

49. *Id.*

50. *Id.*

51. *Id.* at 1520.

52. *Id.* at 1527-28.

2. *The Fair Use Exception in 17 U.S.C. § 107*

In assessing Accolade's fair use defense to Sega's copyright infringement claim, the Ninth Circuit Court of Appeals was guided by the factors set out in § 107 to determine whether a use has been fair. Section 107 sets forth the following four factors in determining whether the use of a work is a fair use:

- (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
- (2) the nature of the copyrighted work;
- (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
- (4) the effect of the use upon the potential market for or value of the copyrighted work.⁵³

In consideration of the first factor, Sega argued that Accolade's sole intent in copying their code was for the commercial purpose of developing software to compete with Sega's software and, therefore, the use was not a fair one.⁵⁴ Although proof of a commercial purpose is normally damaging against the copyright infringer,⁵⁵ the Ninth Circuit Court of Appeals agreed with Accolade that the direct purpose of copying the code was to ascertain the proper functional elements needed to produce software that would be compatible with the Genesis III system.⁵⁶ The court pointed out that the functional aspects of the Genesis III compatible games were not protected by copyright law.⁵⁷ Additionally, the court noted that public policy encourages competition within the industry; therefore, copying a program for the purpose of obtaining a code for compatibility purposes was valid.⁵⁸ This reasoning by the court was groundbreaking. Courts deciding similar cases had previously held that copying software for the purpose of producing other compatible software is not a legitimate reason for infringing copyright law.⁵⁹

In analyzing the second factor of the fair use defense, the court also found in favor of Accolade.⁶⁰ The court explained the difficulty of analyzing the nature of the copyrighted work with re-

53. *Sega*, 977 F.2d at 1521-22 (citing U.S.C. § 107).

54. *Id.* at 1522.

55. *Id.*

56. *Id.*

57. *Id.*

58. *Id.* at 1523.

59. *See generally* *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (1983) (refusing to give the public policy consideration of healthy market competition such credence).

60. *Sega*, 977 F.2d at 1526.

gard to computer software. It indicated that most literary works are in a human-readable form, while the ideas and functions of a computer program cannot be discerned without disassembling the program.⁶¹ In this case, the code needed for compatibility purposes was hidden within Sega's software. The public policy goal of market competition is to be fostered, the court explained, and "[i]f disassembly of copyrighted object code is per se an unfair use, the owner of the copyright gains a de facto monopoly over the functional aspects of his work—aspects that were expressly denied copyright protection by Congress."⁶² The court suggested that "[i]n order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator of the work must satisfy the more stringent standards imposed by the patent laws."⁶³ The court suggested that computer software should be afforded a lower degree of protection than traditional literary works because computer programs contain functional aspects that cannot be protected by copyright law and because the only way to examine the functions of a program is to copy them.⁶⁴

The third factor of the fair use defense, the amount of the original work copied, overwhelmingly favored Sega since Accolade had copied the entire code.⁶⁵ However, the court felt this factor should receive little weight because Accolade had copied the code for the purpose of understanding its function, not to produce new software based only on this portion of the copyrighted program.⁶⁶ Overall, the court decided this factor did not preclude application of the fair use defense.⁶⁷

In assessing the fourth and final factor of the fair use analysis, the Ninth Circuit Court of Appeals took into account that most consumers who own video game systems, such as the Genesis III system, purchase numerous games.⁶⁸ This was an important aspect in this case because if a competitive product is derived from a copyrighted work, and the new product severely diminishes the market value and sales of the original product, the probability of succeeding with a fair use defense is slim. In *Sega*, however, the court referred once again to the public policy of encouraging the free exchange of ideas and market competition and noted that re-

61. *Id.*

62. *Id.* The court cited 17 U.S.C. § 102(b) for this proposition. *Id.*

63. *Id.* In *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 159-64 (1989), the Supreme Court utilized the same reasoning.

64. *Sega*, 977 F.2d at 1526.

65. *Id.*

66. *Id.* See *Sony Corp. of Am. Inc. v. Universal City Studios, Inc.*, 464 U.S. 417, 448-49 (1984) and *Hustler Magazine, Inc. v. Moral Majority, Inc.*, 796 F.2d 1148, 1155 (9th Cir. 1986) for similar reasoning with regard to the third factor of the fair use analysis.

67. *Sega*, 977 F.2d at 1526.

68. *Id.* at 1523.

verse engineering of software encouraged this activity.⁶⁹ Because a new product will normally reduce the sales of an existing product, arguing that a decrease in sales should warrant a finding of unfair use oversimplifies the true intent of the Copyright Act. Simply stated, the courts favor competition, not monopoly. Thus, the court found this factor in favor of Accolade as well.⁷⁰ The importance of the *Sega* court's holding is succinctly represented in the following portion of its opinion. "[W]here disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law."⁷¹

B. Atari Games Corp. v. Nintendo of America, Inc.

It is no coincidence that the holding of *Atari Games Corp. v. Nintendo of America, Inc.*⁷² and the decision in the *Sega* case are similar. *Atari* was decided shortly before the *Sega* Court issued its opinion, and thus established the framework and rationale for the *Sega* decision.⁷³ Like *Sega v. Accolade*, this case arose in the Ninth Circuit, but because it involved a violation of patent law, the Court of Appeals for the Federal Circuit had jurisdiction over the appeal.⁷⁴ In this case, Nintendo of America (Nintendo) filed a lawsuit against Atari Games Corporation (Atari) for copyright and patent infringement.⁷⁵ Nintendo was a manufacturer of a video game system called the Nintendo Entertainment System (NES) and Atari wanted to produce compatible game cartridges for use with the NES.⁷⁶ Nintendo had designed the NES so that the video game system contained a complex electronic key and lock type system that only enabled game cartridges with this specific electronic key to function on the NES.⁷⁷ To obtain this key or object code, a game manufacturer had to set up a licensing agreement with Nintendo and purchase special cartridges with the electronic key from Nin-

69. *Id.* The Ninth Circuit Court of Appeals stated, "[i]t is precisely this growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote." *Id.*

70. *Id.* at 1523-24.

71. *Id.* at 1527-28.

72. 975 F.2d 832 (Fed. Cir. 1992).

73. In its decision, the *Sega* Court cited to the *Atari* opinion. *Sega*, 977 F.2d at 1514 n.1.

74. The Federal Circuit Court of Appeals has jurisdiction over all patent infringement claims under the direction of 28 U.S.C. § 1292 (1994), 28 U.S.C. § 1295 (1994) and 28 U.S.C. § 1338 (1994).

75. *Atari*, 975 F.2d at 835.

76. *Id.* at 835-38.

77. *Id.* at 836.

tendo at a substantial cost.⁷⁸ If a game was not on one of these special cartridges, the game would not run on the NES.⁷⁹

Atari attempted to reverse engineer one of these cartridges in an attempt to decipher the code for the key-lock system.⁸⁰ Atari tried "peeling" the computer chip that contained the code. Peeling is a process whereby engineers gradually peel back a computer chip's layers and microscopically examine the ones and zeros that make up the object code.⁸¹ This process is very costly, time consuming and rarely produces meaningful results.⁸² The peeling procedure failed to produce the code. Atari then monitored the electronic signals passed back and forth between the NES and the game cartridge in an attempt to discover a pattern that might reveal the code. This method also proved unsuccessful.

Atari then fraudulently obtained a copy of the NES's source code from the Copyright Office after misrepresenting that it needed the code for copyright litigation purposes between Atari and Nintendo.⁸³ Atari used the source code to develop a program that unlocked the NES and allowed Atari's games to run on the system.⁸⁴ After Nintendo filed suit against Atari for copyright infringement, the district court entered a preliminary injunction against Atari.⁸⁵

On appeal, Atari argued that it had used a different micro-processor, different instructions and a completely different computer language when designing its program and, as a result, there could be no copyright infringement. Atari also asserted the fair use defense. The court stated that "[a]n author cannot acquire patent-like protection by putting an idea, process, or method of operation in an unintelligible format and asserting copyright infringement against those who try to understand that idea, process, or method of operation."⁸⁶ The court continued its support of the fair use defense arguing that the Copyright Act permits an individual in rightful possession of a copy of a work to undertake necessary efforts understand the work's ideas, processes and methods of operation.⁸⁷ This permission appears in the fair use exception to

78. *Id.*

79. *Id.*

80. *Id.*

81. *Atari*, 975 F.2d at 836.

82. *See Davis*, *supra* note 20 (discussing this peeling technique in greater detail).

83. *Atari*, 975 F.2d at 836. Although the corporations were not in a legal battle at that time, it is ironic that this act of obtaining the code was the source of the later litigation.

84. *Id.* Atari named its comparable program the Rabbit. *Id.*

85. *Id.* at 837.

86. *Id.* at 842.

87. *Id.*

copyright exclusivity.⁸⁸

However, because Atari obtained a copy of Nintendo's source code by illegal means, the court declined to grant Atari its fair use defense.⁸⁹ In order to access the fair use defense and become entitled to its protection, one must have "clean hands."⁹⁰ In other words, anyone asserting the defense must show that it acquired a copy of the program or code by legally authorized means.⁹¹ In this case, Atari probably would have won the case had it obtained the code by reverse engineering lawfully obtained game cartridges.⁹² Unlike the *Sega* court, the *Atari* court did not assess, in any great detail, the four factors to be considered in the fair use defense.⁹³ What is clear, however, is that the *Atari* Court wholeheartedly supported the fair use defense in computer software copyright infringement cases in order to give the public greater access to the program's underlying functions and ideas and promote the advancement of technology. This opinion ultimately was very important as it set forth the reasoning that the *Sega* Court adopted in its opinion.

IV. THE FAIR USE STATUTORY DEFENSE

A. *History and the Changing Law*

While the *Sega* and *Atari* decisions left many aspects of computer software copyright infringement unclear, they did make enormous advances in this area of the law.⁹⁴ The four factors used to determine whether certain reverse engineering efforts should be protected by the fair use defense are not perfectly clear and one court may interpret the results of applying the factors very differently from another court. Many commentators question this case-by-case analysis and ponder whether the United States should develop a *sui generis* system, similar to that adopted in the European Community for deciding computer copyright infringement cases.⁹⁵ However, since the four-step analysis provides a more certain

88. *Id.*

89. *Atari*, 975 F.2d at 843, 846.

90. *Id.* at 846.

91. *Id.*

92. *See Atari*, 975 F.2d at 832 (reasoning that copyright is not exclusive protection for authors).

93. *Compare Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1520-28 (9th Cir. 1992), *with Atari*, 975 F.2d at 843-44, 46.

94. *See generally* John T. Soma, Gus Winfield & Letty Friesen, *Software Interoperability and Reverse Engineering*, 20 RUTGERS COMPUTER & TECH. L.J. 189 (1994) (summarizing some of the unresolved questions left by the *Atari* and *Sega* opinions, including whether it is permissible to use interim copies to reverse engineer software to develop a clone product).

95. *See supra* note 19 (describing the *sui generis* method of software protection).

framework for deciding future cases, it is certainly a move in the right direction.

B. What Do We Know About How to Properly Reverse Engineer?

Until some sort of legislative scheme is developed to deal with the discrepancies in our current system, how should an attorney advise clients who are contemplating reverse engineering a product? It is prudent to recognize several caveats when undertaking a reverse engineering project. First, an attorney should advise his client to obtain an authorized copy of the competitor's software. One of the hurdles that prevented Atari from successfully asserting the fair use defense to its software infringement charge was the fact that it had obtained a copy of Nintendo's source code by misrepresenting to the Copyright Office that it needed the code for litigation.⁹⁶ The Federal Circuit advised others that "[t]o invoke the fair use exception, an individual must possess an authorized copy of a literary work."⁹⁷ To avoid tainting a fair use defense, a computer company should obtain the software on the open market, just as any other consumer would. Additionally, one must be careful not to violate any licensing agreements signed directly with a computer company. For example, when a company such as Sega licenses other software companies to use its code in order to produce compatible games for use on the Sega video game system, the company often must promise not to attempt to reverse engineer any of Sega's codes for use in other games. Courts usually uphold such agreements and will probably view any reverse engineering in these circumstances as tainting a later fair use defense.⁹⁸ The recent trend in court decisions is to accept the fair use defense and allow computer companies to reverse engineer software. However, courts have refused to do so at the expense of more important public policies - avoiding fraud and misrepresentation while en-

96. *Atari*, 975 F.2d at 843. The court noted that "[b]ecause Atari was not in authorized possession of the Copyright Office copy of 10NES, any copying or derivative copying of 10NES source code from the Copyright Office does not qualify as a fair use." *Id.*

97. *Id.* at 843. See also *Harper & Row v. Nation Enters.*, 471 U.S. 539, 562-63 (noting that knowing exploitation of purloined manuscript was not compatible with the "good faith" and "fair dealings" underpinnings of the fair use doctrine).

98. It is also important to note that some commentators have questioned whether shrink-wrap licensing agreements that accompany new software packages could prevent one from reverse engineering the software purchased. Despite conflicting viewpoints, one case suggests that shrink-wrap license agreements may be unenforceable. See Gary W. Hamilton and Jeffrey C. Hood, *The Shrink-Wrap License—Is it Really Necessary?*, 10 *THE COMPUTER L.* 16 (August 1993) (providing additional commentary on the enforceability of shrink-wrap). See also *Vault Corp. v. Quaid Software, Ltd.*, 847 F.2d 255, 270 (5th Cir. 1988) (holding shrink-wrap licensing unenforceable).

couraging fair play.

Second, advise clients to use reverse engineering as a last resort. The *Sega* court specifically concluded "that where disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program . . . disassembly is a fair use of the copyrighted work, as a matter of law."⁹⁹ Indeed, some companies publish certain codes to allow for the development of compatible software. Therefore, if the source code for a particular function or process is obtainable by any other legal means, it should be explored before reverse engineering is undertaken.

Third, if reverse engineering is used out of necessity, engineers should be advised to reverse engineer only those portions of the original program required to decipher the precise functional elements needed for the new program. Courts are weary of reverse engineering efforts that utilize portions of the original program to the extent that the newly developed software is similar in expression to the original. In *Lotus Development Corp. v. Paperback Software International*,¹⁰⁰ the court held that Paperback infringed on Lotus' copyrighted software by devising their spreadsheet software to have a similar look and feel as Lotus' popular 1-2-3 spreadsheet program. Rather than copy only the functional elements of the 1-2-3 program, Paperback had copied much of Lotus' aesthetic qualities as well.¹⁰¹ While it is difficult to say when a court would consider two programs to be too much alike, the *Sega* court attempted to illustrate when reverse engineering may result in the copying of more than the necessary portions of software. Unfortunately, however, these examples are somewhat vague and cannot be readily applied to everyday situations with confidence.¹⁰² To be on the safe side, be sure to avoid copying any program components that relate to the expression of the program or its aesthetic qualities.

Fourth, computer companies should be sure to divide their reverse engineering efforts into two groups of engineers and programmers - one group to reverse engineer the computer program and the other group to develop the new software. This method en-

99. *Sega*, 977 F.2d at 1527-28.

100. 740 F. Supp. 37, 63 (D. Mass. 1990).

101. *Id.* at 69-70. After its success against Paperback, Lotus also brought a copyright infringement suit against Borland International for copying Lotus' 1-2-3 menu commands and portions of its user interface so as to infringe upon the expressional aspect of 1-2-3. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 78, 80 (D. Mass. 1992).

102. David L. Hayes, *The Legality of Disassembly of Computer Programs*, 14 *COMPUTER L.J.* 1, 14 (1992). See also *Whelan Assocs., Inc. v. Jaslow Dental Labs., Inc.*, 797 F.2d 1222 (3d Cir. 1986) (devising an approach to differentiate protected expression from unprotected functions and ideas. This approach, however, has been roundly criticized).

sure clean hands. If a software developer is later charged with software copyright infringement, the company can produce records to show that the newly developed program was completed without directly copying the original software code. Specifically, this process begins with the first group of programmers reverse engineering the original software into the source code, so that human programmers can read it. Next, the first group will explain, in a written journal or log, the functions of the original program as well as the ideas the program uses without describing the expressive content (how the program will look to the user). This journal is then given to the second group of programmers who attempt to design a computer program that emulates the functions and ideas. The second group cannot communicate directly with the original group of engineers and, as a result, helps to insulate the development process from any direct copying of the original program's code. This process should ensure that the end product will look somewhat different than the original program that was reverse engineered. The new program can then be marketed directly against the original program with little fear of violating the original's copyright protection.¹⁰³ Fifth, despite adherence to the previous precautions, computer software companies should take steps to conduct research on the product to be reverse engineered to ensure that patent law does not provide protection for the particular process or function that is to be reverse engineered and used in a new program. Keep in mind that while copyright law cannot protect functions and ideas, patent law does.¹⁰⁴ Because of the rigorous requirements that must be met in order to get a patent on certain functions or processes, very few computer programs are patented. Nevertheless, it is important to protect yourself from any patent infringement lawsuits by inquiring into this area before reverse engineering and marketing a new program.

Lastly, a prudent software developer should consult with an intellectual property attorney to be sure that its reverse engineering efforts are in compliance with the law. In this rapidly changing area of law, new cases are being decided everyday. With the uncertainty that exists currently, the law tends to change drastically in short amounts of time.

V. CONCLUSION

Justice O'Connor suggested that there is a major public benefit to reverse engineering in that it encourages inventors and busi-

103. See Philip J. McCabe, *Reverse Engineering of Computer Software: A Trap For the Unwary?*, 9 THE COMPUTER L. ASS'N BULL. 1 (1994), No. 2, for additional support of this procedural method.

104. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839 (Fed Cir. 1992); McCabe, *supra* note 103, at 11.

nesses to devise similar products.¹⁰⁵ Marketing similar products serves to encourage the original product's inventors to continue to improve the product in order to stay ahead of the competition. It also encourages healthy competition and price pressures in the open market.¹⁰⁶ It appears as though recent cases like *Sega* and *Atari* adopt this same viewpoint. While early case law seemed to lean towards overprotection of software copyrights, the trend now is to promote the progress of science and the useful arts.¹⁰⁷

Favoring free access to program functions and ideas not protected by patent law can only serve to promote further creativity in a rapidly progressing industry. As a result, courts should give more credence to the fair use defense in order to enable United States computer companies to compete on a level playing field with overseas competitors, who are subject to much less restrictive regulations.

While the *Sega* and *Atari* decisions made great strides in developing the law in this area, more clarity is needed so that computer software manufacturers have a better idea of what they can and cannot do when developing new programs. Whether this takes the form of new legislation or more potent case law, it is clear that more guidance is needed.

For those companies wishing to safeguard newly developed software, it appears that patent law may be the best protection. As mentioned earlier, however, successfully receiving a software patent entails passing through a rigorous maze of formalities and leaping through many complex hoops which may render this protection infeasible when weighing the time, money, and effort against its benefits. In many instances, by the time a patent is received on software, the software is already outdated. Conversely, protecting software with copyright law can also be inadequate, as evidenced by the emerging trend of case law already discussed.

Perhaps the most favorable way of protecting software without sacrificing the benefits of access to new technology in this scientific field is to adopt a *sui generis* system much like the one adopted by the European Community. A *sui generis* system could combine the low cost and speed of copyright law with the protection of patent law. By balancing the favorable characteristics of copyright and patent law, software companies would still be able to reap the rewards of producing new technology without worrying about having their programs quickly copied by competitors. Society, however, would still benefit by eventually having access to the new processes and techniques utilized in new software.

105. *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 160 (1989).

106. *Id.*

107. U.S. CONST. art. I, § 8, cl. 8.

The law as it applies to this field is evolving slowly while the computer changes drastically in very brief time periods. Ultimately, the law will have to catch up to be useful to those in the industry. The only way to accomplish this is to adopt a *sui generis* system specifically tailored to this area of law so that all competitors in the industry will know the rules before they play the game.