

SONY COMPUTER ENTERTAINMENT, INC. V. CONNECTIX CORP.

By Stan Karas

Software is a utilitarian article.¹ Copyright law, on the other hand, is commonly associated with protecting creative expression in such fields as music, literature, and film.² Yet, software is protected under the law of copyright.³ Reconciling software's utilitarian nature with its copyrightability has presented unique challenges for the courts.⁴ One of the most interesting challenges has been ensuring public access to software's uncopyrightable functional⁵ elements. Culminating in the Ninth Circuit's recent decision in *Sony Computer Entertainment, Inc. v. Connectix Corp.*,⁶ a line of cases has held that reverse engineering of software to gain access to its functional elements is fair use.

In *Sony v. Connectix*, the Ninth Circuit concluded that Connectix's reverse engineering⁷ of the an operating system program extracted from a

© 2001 Regents of the University of California.

1. "Software" is a collective term for "computer programs and applications . . . that can be run on a particular computer system." MICROSOFT ENCARTA WORLD ENGLISH DICTIONARY (Kathy Rooney et al. eds. 1999), available at <http://dictionary.msn.com/find/entry.asp?search=software> (last visited Feb. 7, 2001). "Computer software" thus includes operating systems, compilers, interpreters, and application programs.

2. "The overarching object of copyright law in the United States is to encourage the widest possible production and dissemination of literary, musical, and artistic works." PAUL GOLDSTEIN, COPYRIGHT: PRINCIPLES, LAW, AND PRACTICE § 1.1 (1989).

3. The 1980 amendment to the Copyright Act amended 17 U.S.C. § 117 to provide that "it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program." 17 U.S.C. § 117 (1994); see also H.R. REP. NO. 94-1476, at 54 (1976) (explaining that § 102(a)'s reference to "literary works" "includes . . . computer programs"); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983) (programs in machine-readable form are appropriate subject matter for copyright).

4. See *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992) ("Computer programs pose unique problems for the application of the 'idea/expression distinction' that determines the extent of copyright protection.").

5. Copyright law does not protect ideas contained in a work. See Part II. A *infra*. As used in this Note, the term "functional element" refers to such uncopyrightable material.

6. 203 F.3d 596 (9th Cir. 2000), *cert. denied*, 2000 U.S. LEXIS 5843 (2000).

7. As used in this Note, unless otherwise noted, the term "reverse engineering" refers only to reverse engineering of software.

Sony PlayStation console was protected fair use.⁸ Connectix's repeated copying of copyrighted elements of Sony's software did not constitute infringement because it was necessary to access the software's functional elements.⁹ This Note explains that while the court's application of the fair use doctrine is consistent with precedent, the decision is significant because it permits reverse engineering of software in the process of creating a product that will compete directly with the original. This outcome conflicts with the "effect on potential market" factor of the fair use test. The Note will also discuss the possible consequences of *Sony v. Connectix*, including software manufacturers' greater reliance on software patents and shrinkwrap licenses, as well as significant changes in business and marketing practices of software manufacturers.

I. BACKGROUND

A. Copyright Law and the Fair Use Doctrine

The Constitution grants Congress the power to enact copyright legislation.¹⁰ Copyright law confers on the copyrighted work's author a limited statutory monopoly through the grant of certain exclusive rights.¹¹ The copyright owner's exclusive rights are limited in a number of ways, including by the fair use limitation.¹² The fair use doctrine permits the courts to avoid finding copyright infringement when doing so "would stifle the

8. *Connectix*, 203 F.3d at 599.

9. *Id.*

10. U.S. CONST. art. I, § 8, cl. 8 ("The Congress shall have the power . . . 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; . . .").

11. The copyright owner has the exclusive rights to do and to authorize any of the following:

(1) to reproduce the copyrighted work in copies or phonorecords; (2) to prepare derivative works based upon the copyrighted work; (3) to distribute copies or phonorecords of the copyrighted work to the public; (4) to perform the copyrighted work publicly; (5) to display the copyrighted work publicly; and (6) in the case of sound recordings, to perform the copyrighted work publicly by means of a digital audio transmission.

17 U.S.C. § 106 (1994 & Supp. 1996). This Note's focus is on the first of these rights—reproduction of the copyrighted work in copies.

12. 17 U.S.C. § 107 (1994). See, e.g., Dane S. Ciolino, *Rethinking the Compatibility of Moral Rights and Fair Use*, 54 WASH. & LEE L. REV. 33, 56 ("An integral part of the Copyright Act, fair use limits copyright's protection of intangible intellectual works of authorship.").

very creativity which that law is designed to foster.”¹³ Under section 107 of the Copyright Act, fair use analysis includes the following factors:

- (1) the purpose and character of the use, including whether the use is of a commercial nature;
- (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.¹⁴

These factors are weighed together in determining whether copying is protected by fair use.¹⁵ Additionally, to encourage creativity and the free exchange of ideas, copyright protection does not extend to the ideas embodied in a work, but only to the work’s expression.¹⁶

B. Principles of Reverse Engineering

Reverse engineering is the process of disassembling a finished product to uncover its functional components. In the context of trade secret law, the Supreme Court has endorsed reverse engineering as a “fair and honest means [of] . . . starting with the known product and working backward to divine the process which aided in its development or manufacture.”¹⁷ With regard to software, reverse engineering involves decompiling the object code in which software is distributed and translating it into source code,

13. See *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 577 (1994).

14. 17 U.S.C. § 107 (1994).

15. *Campbell*, 510 U.S. at 578.

16. See 17 U.S.C. § 102(b) (1994) (The Act does not protect “any idea, procedure, process, system, method of operation, [or] concept . . .”).

Because it contains both functional and protectable elements, software is markedly different from such conventional subjects of copyright protection as literary works. This unique position of computer code within the range of copyrightable material requires a unique approach to the determination of infringement. In *Computer Associates International v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), the Second Circuit adopted a three step test for determining copyright infringement of non-literal elements of a computer program. First, a court should apply an “abstraction test” by “dissect[ing] the allegedly copied program’s structure and [isolating] each level of abstraction contained within it.” *Id.* at 707. Second, a court should apply a “filtration test” by examining the structural components at each level of abstraction to “determine whether their particular inclusion at that level was ‘idea’ or was dictated by considerations of efficiency,” which would make it merely incidental to that idea; “required by factors external to the program itself; or taken from the public domain.” *Id.* Finally, the court should compare the allegedly infringing work with the “protectable expression” found in the filtration step. *Id.* at 710-11.

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

which programmers can then use to understand the functional elements of the program.¹⁸

Reverse engineering of software may take several forms: (1) reading about the program; (2) observing "the program in operation by using it on a computer;" (3) performing a "static examination of the individual computer instructions contained within the program;" and (4) performing a "dynamic examination of the individual computer instructions as the program is being run on a computer."¹⁹ The first method is often ineffective because information available about a program may be scarce and the existing manuals may misdescribe a program.²⁰ The remaining methods require that the copyrighted program be copied into the computer's random access memory ("RAM").²¹ Such copying, if unauthorized, has been found to be actionable under the Copyright Act.²² The crucial issue is whether copying for the purpose of deciphering functional elements of the program falls within the fair use defense.

C. Application of Fair Use Defense in Software Copyright Infringement Actions

When confronting copyright infringement stemming from reverse engineering of software, most courts have held that such copying constitutes fair use. For example, in *Atari Games Corp. v. Nintendo of America, Inc.*,²³ the Federal Circuit excused such copying as fair use when it was necessary to obtain access to the software's functional elements.²⁴ This case involved Nintendo's 10NES software, which contained a security mechanism requiring the console to detect a coded message on the video game cartridge in order to function. Atari, a video game manufacturer, used a copy of Nintendo's program to correct errors in its reverse engineering efforts and subsequently created its own software, the "Rabbit

18. Greg Weiner, *Reverse Engineering as a Method of Achieving Compatibility in the Computer Industry*, 6 U. BALTIMORE INTELL. PROP. J. 1, 2 (1997).

19. *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596, 599 (9th Cir. 2000) (citing Andrew Johnson-Laird, *Software Reverse Engineering in the Real World*, 19 U. DAYTON L. REV. 843, 846 (1994)).

20. *Connectix*, 203 F.3d at 600.

21. *Id.*

22. Merely loading software into the RAM of a computer has been held to create a copy within the meaning of the Copyright Act. *See, e.g., MAI Sys. Corp. v. Peak Computer, Inc.*, 991 F.2d 511, 519 (9th Cir. 1993).

23. 975 F.2d 832 (Fed. Cir. 1992).

24. This extension of the fair use doctrine was dicta; the Federal Circuit upheld the lower court's preliminary injunction against Atari, on the grounds that Atari acquired an unauthorized reproduction of source code from the Copyright Office, which negated any finding that Atari's reverse engineering was fair use. *Id.* at 841-42.

Program,” to unlock the 10NES system. The result was “to render the Atari Games’ chip functionally indistinguishable from the Nintendo chip, thus precluding Nintendo from altering its future base units in a manner that would selectively exclude Atari Games’ cartridges.”²⁵ In other words, the program allowed Atari to produce video games that were not licensed by Nintendo but could still be played on the Nintendo game system. Atari’s final code differed on a line-by-line basis from that of Nintendo.²⁶ The court held that Atari’s reverse engineering was fair use as long as Atari did not copy beyond the point necessary to understand the unprotected ideas and processes embedded in 10NES software or commercially exploit Nintendo’s protected expression.²⁷ Even though the court ruled against Atari on other grounds, *Atari* established fair use as a valid defense in reverse engineering copyright suits.

Central to the *Connectix* court’s analysis was *Sega Enterprises v. Accolade, Inc.*,²⁸ which was decided only months after *Atari*. In *Sega*, during the process of creating video games to be played on Sega’s console, Accolade copied and disassembled some of Sega’s copyrighted software.²⁹ Accolade then printed and studied the disassembled code in order to produce Sega-compatible games.³⁰ The Ninth Circuit held that creation of object code through disassembly constituted infringement but concluded that intermediate copying, such as Accolade’s, was fair use.³¹

The court admitted that Accolade’s purpose in copying was purely commercial.³² Nevertheless, the copying was fair use because it occurred at an intermediate stage of software development, and therefore Accolade’s commercial exploitation of Sega’s work was indirect.³³ Furthermore, Accolade’s purpose was nonexploitative, since it disassembled Sega’s code merely to study its functional requirements.³⁴

The court also observed that Accolade’s games would not displace Sega’s games in the market and that limiting the market for Sega-compatible games only to those produced or licensed by Sega would run counter to the goal of promoting creative expression embodied in copy-

25. *Atari Games Corp. v. Nintendo of America, Inc.*, 18 U.S.P.Q.2d (BNA) 1935, 1939 (N.D. Cal. 1991).

26. *See Atari Games*, 975 F. 2d at 836.

27. *See id.* at 843.

28. 977 F.2d 1510 (9th Cir. 1993).

29. *Id.* at 1514.

30. *Id.* at 1514-15.

31. *See id.* at 1514.

32. *Id.* at 1522-23.

33. *Id.*

34. *Id.* at 1523.

right law.³⁵ Ultimately, the court held that “[w]here there is good reason for studying or examining the unprotected aspects of a copyrighted computer program, disassembly for purposes of such study or examination constitutes a fair use.”³⁶

Accordingly, prior reverse engineering cases establish these principles: (1) the defendant may only copy the minimum amount necessary to understand the product; (2) the defendant must have a legitimate reason to reverse engineer the software; (3) the defendant must lawfully obtain the copy of the plaintiff’s work; and (4) disassembly must be the only reasonable way to gain access to the ideas contained in the software.

II. CASE SUMMARY

In 1992, Sony Computer Entertainment, Inc. (“Sony”) began the development of the Sony PlayStation, a video game system. The development and marketing of the PlayStation cost Sony over \$600 million and took more than three years.³⁷ Introduced in 1995, the PlayStation became a success, selling over 20 million copies worldwide and establishing Sony as a leader in the video game console industry.³⁸

In July 1998, Connectix, Inc., a software manufacturer, decided to emulate PlayStation’s hardware so that PlayStation games could be played on personal computers.³⁹ To develop its system, Connectix needed to emulate the PlayStation operating system—the basic input-output software (“BIOS”).⁴⁰ Sony had registered a copyright in its BIOS and disallowed copying or distribution for any purpose.⁴¹ To accurately emulate the PlayStation’s BIOS, Connectix purchased a PlayStation, removed the chip containing the BIOS, and downloaded the contents of that chip onto a disk.⁴² Over the several months spent developing a functional PlayStation emulator,⁴³ Connectix engineers repeatedly copied Sony’s BIOS into the

35. *See id.* at 1523-24.

36. *Id.* at 1520.

37. *See Sony Computer Entm’t, Inc. v. Connectix Corp.*, 48 F. Supp. 2d 1212, 1214 (N.D. Cal. 1999).

38. *Id.*

39. *Id.* at 1215.

40. *Id.*

41. *Id.*

42. *See id.* at 1216.

43. An emulator is hardware or software that permits a computer system to run programs written for and process data originating from a different type of computer system. MICROSOFT ENCARTA WORLD ENGLISH DICTIONARY (Kathy Rooney et al. eds. 1999), available at <http://dictionary.msn.com/find/entry.asp?search=emulator> (last visited Feb.

RAM of their personal computers during the process of reverse engineering.⁴⁴ This copying was intermediate; it was done solely to understand the functioning of the PlayStation. Ultimately, none of Sony's copyrighted computer code appeared in Connectix's emulator program.⁴⁵

In January 1999, after spending about \$150,000 on its development, Connectix introduced the Virtual Game System ("VGS") at the MacWorld Expo in San Francisco.⁴⁶ Sony promptly sued Connectix in the District Court for the Northern District of California, alleging copyright infringement as well as other causes of action.⁴⁷

A. District Court Decision

In its April 1999 decision, the district court granted a preliminary injunction prohibiting Connectix from marketing its program. The court held that Connectix had likely infringed the copyrighted code contained within the PlayStation BIOS.⁴⁸ As a result, it ordered Connectix to pull the Virtual Game Station from the market and halt all moves towards preparing its future retail release.⁴⁹

The court rejected Connectix's fair use defense. The court stated that even though the VGS did not contain any infringing code, it was, and had been marketed as, a substitute for the PlayStation.⁵⁰ The fact that Connectix's product allowed PlayStation games to be played on a computer monitor as opposed to a television screen did not amount to transformative use.⁵¹ According to the court, since the VGS was merely a substitute product, marketing it would harm Sony's sales of its PlayStation console.⁵² Accordingly, the district court issued a preliminary injunction against Connectix.

7, 2001). In other words, emulator products are designed to enable one computer system to imitate another, so that they can function in the same way and achieve the same results.

44. Sony's Appellate Brief at 10, *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000) (No. 99-15852).

45. *Connectix*, 203 F.3d at 600.

46. *Id.* at 601.

47. *Id.* Sony also alleged circumvention of technological protection measures, and trademark dilution. *Sony Computer Entm't, Inc. v. Connectix Corp.*, 48 F. Supp. 2d 1212, 1214 (N.D. Cal. 1999).

48. The court also found that the Connectix product had the potential of tarnishing the Sony PlayStation trademark. *Id.* at 1223.

49. *See id.* at 1224.

50. *See id.* at 1219.

51. *See id.* ("The VGS does not do anything new, anything different, or anything unique from the PlayStation.")

52. *Id.* at 1221.

B. Ninth Circuit Decision

On appeal, the Ninth Circuit ruled that copying by Connectix constituted fair use and dissolved the lower court's injunction.⁵³ The court based its decision primarily on *Sega Enterprises v. Accolade, Inc.*,⁵⁴ which held that disassembly is fair use as a matter of law where it is necessary to gain access to the ideas contained within a copyrighted program.⁵⁵ Taking this as a starting point for its analysis, the court found that a fair use analysis favored Connectix.⁵⁶

With respect to the nature of the copyrighted work, the first fair use factor, the court found that because Sony's BIOS program contains unprotectable aspects it did not belong to the core of intended copyright protection as, for example, would a literary work. Under this lower standard of protection, to establish fair use Connectix only had to demonstrate that its copying was "necessary." The court found that this showing had been made. Sony's BIOS contained functional elements that could not be accessed without copying the program during the process of reverse engineering.⁵⁷ The court rejected Sony's argument that Connectix's copying was infringing because it "used" rather than merely "studied" the copyrighted code, as creating an "artificial" distinction.⁵⁸

Responding to Sony's claim that it was not necessary for Connectix to copy the BIOS repeatedly, the court held that the number of times that the program is disassembled is irrelevant to a finding of fair use.⁵⁹ The court noted that given the technical realities of reverse engineering, limiting the number of times the code can be copied leads to inefficiency.⁶⁰ Erecting

53. *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596, 608-09 (9th Cir. 2000). The court also rejected Sony's trademark dilution claims, finding that "the evidence on the record did not support a finding of misattribution." *Id.* at 609. Furthermore, the court could not find any evidence that "Sony's mark or product was regarded or was likely to be regarded negatively because of its performance on Connectix's Virtual Game Station." *Id.*

54. 977 F.2d 1510 (9th Cir. 1992).

55. *See id.* at 1518.

56. *Connectix*, 203 F.3d at 608. The only fair use factor weighing in favor of Sony was the amount and substantiality of portion of the copyrighted material used. *Id.* at 606. Connectix admitted to copying Sony's BIOS multiple times in order to reverse engineer it. *Id.* The court, however, noted that this factor of the test is of little weight when the final product does not contain infringing material. *Id.* at 605.

57. *See id.* at 603.

58. *Id.* at 604.

59. *See id.*

60. *Id.*

artificial hurdles to the public's access to the ideas contained within copyrighted works contravenes the goals of copyright law.⁶¹

Further, since Connectix developed entirely new object code, the court held that its use of copyrighted material was transformative under the "purpose and character of the use" prong of the fair use test.⁶² Accordingly, the court was "at a loss" to see just how Connectix's development efforts, which resulted in a noninfringing product, violated any PlayStation copyrights.⁶³ Even though the PlayStation and the VGS are similar in both function and screen output, the court found that the VGS, as a "wholly new product," was "modestly transformative."⁶⁴

Finally, the court found that the fair use factor concerning the effect of the use on the potential market favored Connectix. While acknowledging that Connectix products may cause a loss of PlayStation sales, the court held that this made Connectix a legitimate competitor and not an infringing party.⁶⁵ Since the VGS is a transformative product, endowed with original expressive qualities, it is less likely to cause a substantial adverse effect on the market for the original.⁶⁶ According to the court, Sony's potential economic losses result from legitimate competition and not Connectix's supplanting or superseding the market for the PlayStation.⁶⁷ Consequently, the Ninth Circuit reversed the district court's ruling.

III. DISCUSSION

Although the Ninth Circuit's decision in *Sony v. Connectix* was mandated by precedent, the court did not devote sufficient attention to the fair use factor concerning the effect of use on the market for the original work. Consequently, the court for the first time found production of an emulator that is functionally identical to the original to be fair use. This holding is likely to encourage greater reliance by software manufacturers on software patents and shrinkwrap licenses prohibiting reverse engineering, and result in improved quality of software and greater market choice for consumers.

61. *Id.*

62. The fact that copying was done for a commercial purpose does not weigh against Connectix, since the purpose in copying the BIOS was legitimate and the commercial use of copyrighted material was only intermediate.

63. *Id.* at 606-07.

64. *Id.*

65. *See id.* at 607

66. *See id.*

67. *See id.*

A. Reverse Engineering and Emphasis on Software's Functional Nature

The *Connectix* court's fair use analysis is predicated on its approach to the "nature of the work" factor. By setting out that the PlayStation BIOS as a software program was entitled to "a lower degree of protection than more traditional literary works,"⁶⁸ the court ruled that in order to be fair use, *Connectix's* copying of Sony's copyrighted code need only be "necessary." Although this approach permits extensive copying of copyrighted code, it is a pragmatic and fair method of facilitating access to software's functional elements.

This permissive approach to reverse engineering is firmly based in precedent. In *Vault Corp. v. Quaid Software Ltd.*,⁶⁹ the Fifth Circuit rejected the plaintiff's argument that reverse engineering a computer program was contrary to the Copyright Act because it was not for the "intended purpose" of executing the program. The court reasoned that such a narrow reading of the Copyright Act was contrary to the Act's language.⁷⁰ Similarly, in *Atari* the Federal Circuit emphasized that reverse engineering is justified because individuals have a right to "undertake necessary efforts to understand the [copyrighted] work's ideas, processes, and methods of operation."⁷¹ Since software is usually distributed to the public only as object code, which is unintelligible to most humans, the holder of the copyright cannot prevent the public from attempting to access the ideas contained within that code via reverse engineering.⁷² The *Atari* court expressly linked the right to reverse engineer software with the weak level of copyright protection afforded to computer code. "When the nature of a work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying."⁷³

68. *Id.* (quoting *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510, 1526 (9th Cir. 1992)).

69. 847 F.2d 255 (5th Cir. 1988).

70. *Id.* at 261.

71. *Atari Games Corp. v. Nintendo of America, Inc.*, 975 F.2d 832, 842 (Fed. Cir. 1992).

72. *See id.* ("An 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. The Copyright Act permits an individual in rightful possession of a copy of a work to undertake necessary efforts to understand the work's ideas, processes, and methods of operation." (citations omitted)).

73. *Id.* at 843.

The *Sega* decision reinforced judicial endorsement of reverse engineering as a legitimate method of getting to the functional elements contained in computer object code. The court emphasized that “computer programs are, in essence, utilitarian articles—articles that accomplish tasks.”⁷⁴ Where the copyrighted work contains little creative expression, copyright protection is thin.⁷⁵ The amount of creative expression in software is limited by its functional purpose; even if programming decisions are idiosyncratic and original, they still serve a functional purpose. Accordingly, the court explained that given the unique nature of software in the copyright regime, copying of an entire program is permissible if it is “the only means of gaining access to . . . unprotected aspects of the program.”⁷⁶

B. Rejection of the Strict Necessity Test

The *Connectix* decision expands *Sega*'s principle of necessity. In *Connectix*, the court rejected Sony's contention that *Sega* limited fair use for copying code to the number of instances strictly necessary to access its functional elements.⁷⁷ The *Connectix* court specified that once the necessity of the defendant's method of accessing software's functional elements is established, the number of times that method is applied is irrelevant.⁷⁸ In other words, as long as Connectix necessarily had to make one copy of the BIOS code in order to study it, it could make and use hundreds of subsequent copies to expedite its reverse engineering.⁷⁹

Although the court maintained that *Sega* mandates this result, that case is factually distinguishable from *Connectix*. Unlike Connectix, Accolade did not spend months copying and running tests on the copyrighted software in order to produce an emulator.⁸⁰ In both cases, decompilation was necessary to access the ideas contained within the software, but only Connectix continued with the copying until the emulator was completed. In

74. *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992).

75. *See id.* (citing *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349 (1991)).

76. *Sega*, 977 F.2d at 1520. The court specifically noted the lack of evidence of a viable alternative to reverse engineering as a factor weighing in favor of finding fair use. *See id.* at 1525-26.

77. *See Sony Computer Entm't, Inc. v. Connectix Corp.* 203 F.3d 596, 604 (9th Cir. 2000).

78. *See id.* at 605.

79. David Goldberg & Robert J. Bernstein, *Computers and Intermediate Copying*, N.Y. L.J., July 21, 2000, at 3.

80. In *Sega*, Accolade disassembled the code, then printed it out and studied its functional elements to produce Genesis-compatible videogames.

Connectix, the Ninth Circuit rejected the opportunity to limit the frequency of copying of copyrighted code to what was necessary to divine the workings of the program, choosing instead a much broader “necessity of method” standard. This is a pragmatic and justifiable choice.

Sony urged the court to consider the frequency of copying as relevant to the issue of whether the conduct was reasonably necessary. Cognizant of the difficulty of applying established legal standards to evolving technologies, the Ninth Circuit refused to “supervise the engineering solutions of software companies in minute detail. . . .”⁸¹ The court’s refusal to create a stricter test for fair use in reverse engineering cases reflects its caution in assessing complex technologies as well as its recognition of the judiciary’s inability to create lasting legal tests for rapidly developing industries.⁸²

Indeed, limiting the extent of permissible copying to what is reasonably necessary would have caused confusion in the lower courts. If a “strict necessity” standard were adopted, trial judges in subsequent cases would have to face battles of experts on whether a particular instance of intermediate copying was necessary to access a program’s functional elements. Given the inherent subjectivity at this level of inquiry, further refinement of the strict necessity standard would have introduced substantial uncertainty into this area of the law. Embracing a strict necessity standard would require the courts to issue highly arbitrary and potentially uneducated decisions. This arbitrariness would impede creativity by evoking the specter of costly litigation at the early stages of the reverse engineering process.

The lower level of copyright protection afforded to software permits any unauthorized copying of software necessary to access the ideas contained therein. All reverse-engineering-related copying is necessary per se because object code is not readable by humans. Instead of expanding the legal test of necessity by considering the frequency of copying, the court pragmatically chose to permit any amount of intermediate copying so long as the other factors of the fair use test are satisfied. This decision is within the spirit of precedent, reflects the technical and market realities of software engineering, and comports with the institutional competence of the courts.

81. *Connectix*, 203 F.3d at 605.

82. As the *Sega* court noted, the courts have not been able to develop even a rudimentary test standard distinguishing between functional and expressive elements in software. The court observed that “thus far, many of the decisions in this area reflect the courts’ attempt to fit the proverbial square peg in a round hole.” *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992) (quoting *Computer Assocs. Int’l, Inc. v. Altai, Inc.* 982 F.2d 693, 712 (2d Cir. 1992)).

C. Emulators and the Fourth Fair Use Factor

Sony v. Connectix further extends *Sega* and other reverse engineering cases by finding the development of an emulator through reverse engineering to be fair use. Although the court's decision that the "nature of the work" factor of the fair use test favored Connectix is justifiable, its finding in favor of Connectix on the "effect of use upon the potential market" factor is questionable. The Ninth Circuit's decision encourages production of emulator products through reverse engineering,⁸³ which will have a deleterious effect on the market for the originals. This result conflicts with the court's analysis of the "effect of use" factor.

The Ninth Circuit was faced with an issue of first impression: Is an emulator whose code does not infringe the original but that performs the same functions protectable under the fair use defense? In *Atari*, the "Rabbit Program" did not serve as a market substitute for Nintendo Entertainment System; it simply gave Atari a way to unlock Nintendo's software so that Atari could produce Nintendo-compatible games. In *Sega*, Accolade created original video games to be played on Sega's console. The court found that although Accolade's games may affect the market for Sega's or Sega-licensed video games, there is no proximate link between the new product and a reduction of sales of the original.⁸⁴ Accolade's games did not supplant Sega's; they added to the marketplace. No other case has dealt with emulator products.⁸⁵

83. As one booster for the emulation scene put it: "There's no more questioning the legality of emulation. *Sony v. Connectix* finally settles that issue once and for all. Both developing and using an emulator are protected under the 'fair use' statute of copyright law." David Lloyd, *Y2K Addendum: The Effects Sony v. Connectix - Analysis and Implications*, at http://www.eidolons-inn.de/emufaq2000/EmuFAQ_Y2KAddendum1.htm (Mar. 16, 2000).

84. By facilitating the entry of a new competitor, the first lawful one that is not a Sega licensee, Accolade's disassembly of Sega's software undoubtedly "affected" the market for Genesis-compatible games in an indirect fashion. We note, however, that . . . video game users typically purchase more than one game. There is no basis for assuming that Accolade's "Ishido" has significantly affected the market for Sega's "Altered Beast," since a consumer might easily purchase both; nor does it seem unlikely that a consumer particularly interested in sports might purchase both Accolade's "Mike Ditka Power Football" and Sega's "Joe Montana Football," particularly if the games are, as Accolade contends, not substantially similar.

Sega, 977 F.2d at 1523.

85. See, e.g., *DSC Communications Corp. v. DGI Techs., Inc.*, 81 F.3d 597, 601 (5th Cir. 1996); *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532, 1539 n.18 (11th Cir. 1996);

Emulation presents novel legal challenges that the *Connectix* court skirted over. The Ninth Circuit dispensed with the “effect of use on potential market” factor in three terse paragraphs.⁸⁶ The court’s analysis of the factor hinged on its earlier finding that the VGS was a transformative work.⁸⁷ The court assumed that because the VGS was transformative, it was less likely to adversely affect the market for the PlayStation.⁸⁸ The transformative nature of the VGS also made it a “legitimate competitor in the market for platforms on which Sony and Sony-licensed games can be played.”⁸⁹ Consequently, “some economic loss by Sony as result of this competition does not compel a finding of no fair use.”⁹⁰

Yet, the finding of “modest” transformativeness does not compel a finding of fair use under the fourth factor. As the court itself noted, although a transformative work is less likely to cause a substantially adverse impact on the potential market for the original, it may still do so.⁹¹ Where the allegedly infringing product is used for the same intrinsic purpose as the original, that fact cuts against a finding of fair use.⁹² This is the case here, and the court should have addressed the functional identity of the PlayStation and the VGS.

The VGS performs the same function as the PlayStation: providing a console for running PlayStation games. Emulators are by definition a replacement for the original product, not a “supplement” as the court saw it. The Ninth Circuit recognized that the VGS would cause Sony to lose console sales and profits.⁹³ The district court was blunter: “Sony is being harmed by the sales of the *Connectix* emulator. To the extent an individual

Lotus Dev. Corp. v. Borland Int’l, Inc. 49 F.3d 807, 817-18 (1st Cir. 1995) (Boudin, J., concurring); Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 270 (5th Cir. 1988).

86. *Connectix*, 203 F.3d at 607-08.

87. “Merely stating that the final work was transformative was all the proof the *Connectix* court needed to hold that [the ‘effect of the use’] factor was in favor of *Connectix*.” Morgan Malino, *Focus on Copyright*, THE NATIONAL LAW JOURNAL, Oct. 16, 2000, at C11.

88. *See Connectix*, 203 F.3d at 607.

89. *Id.*

90. *Id.*

91. *See id.*

92. *See Am. Geophysical Union v. Texaco, Inc.*, 60 F.3d 913, 922 (2d Cir. 1994) (The court rejected the fair use defense where copyrighted works were being photocopied without paying for a license and noted that “courts will not sustain a claimed defense of fair use when the secondary use can fairly be characterized as a form of ‘commercial exploitation,’ i.e., when the copier directly and exclusively acquires conspicuous financial rewards from its use of the copyrighted material.”); *see also Marcus v. Rowley*, 695 F.2d 1171, 1175 (9th Cir., 1983).

93. *Connectix*, 203 F. 3d at 607.

purchases a VGS to play PlayStation games, those consumers [sic] will be less likely to buy PlayStation consoles.”⁹⁴ Indeed, the only reason for purchasing both consoles would be to play PlayStation video games on both a personal computer (“PC”) and a television. Given the high prices of consoles, such a duplicative purchase is unlikely. Consequently, since the VGS is used for the same intrinsic purpose as the PlayStation, it is at best only mildly transformative.⁹⁵

Even if we assume that the VGS is entirely transformative, the detrimental effect on Sony’s market for its product may be extensive. The VGS reduces Sony’s prospects of expanding its product line by creating a PC-based PlayStation or licensing its BIOS to another company for that purpose.⁹⁶ More generally, the *Connectix* decision makes long-term product development or licensing planning by the copyright holders extremely difficult. For example, at any point during licensing negotiations a competitor may unexpectedly come out with an emulator and render moot a potentially lucrative agreement. The effect on the potential market for the original is substantial and harmful.

Also, there is the possibility of Connectix patenting its product to preclude Sony from entering the market for PC-based PlayStation emulators. Although Sony has valid patents on its BIOS, another company in a similar situation may not. As the following section explains, this situation may lead to greater reliance on software patents by copyright holders.

D. Possible Consequences of *Sony v. Connectix*

It is hard to predict how a single court decision will affect an industry as complex and as rapidly developing as software manufacturing. It is, however, possible to perceive several probable effects of *Sony v. Connectix*.

94. *Sony Computer Entm’t, Inc. v. Connectix Corp.*, 48 F. Supp. 2d 1212, 1221 (N.D. Cal 1999).

95. Admittedly, on a line-by-line basis the VGS’s code is different from the PlayStation’s. As seen in *Atari* and *Sega*, this often compels a finding of fair use. Transformative use entails adding something new or original to the existing work. *Campbell v Acuff-Rose Music, Inc.*, 510 U.S. 569, 579 (1994). *Connectix* reworked Sony’s software so that PlayStation games could be played on a new platform. As a result, the VGS adds to the marketplace by expanding the number of platforms for PlayStation games. This is, however, undercut by the functional identity of the two products.

96. There is also no suggestion on the record that Sony was not already working on a PC-compatible version of the PlayStation when the VGS hit the market.

1. *Greater Use of Software Patents*

One likely outcome of the Ninth Circuit decision is increased reliance by software manufacturers on software patents.⁹⁷ Since Sony did not sue Connectix for patent infringement, the *Connectix* court noted that its decision did not affect patent claims.⁹⁸ Less than a week after the Ninth Circuit announced its decision in *Connectix*, Sony filed a new suit against Connectix alleging patent infringement.⁹⁹ Similarly, Sony filed a patent suit against Bleem, Inc., another emulator manufacturer, after losing a copyright infringement suit against that company over Bleem's use of screen shots from Sony's video games in advertisements for its emulator.¹⁰⁰ Therefore, software manufacturers' attempt to protect their products from reverse engineering by relying on software patents is not an abstract possibility but a reality.¹⁰¹

As opposed to copyright law, under patent law reverse engineering of patented software likely constitutes infringement.¹⁰² Therefore, the public is unable to access the ideas contained within patented software by reverse engineering it. Furthermore, the public may be unable to gain that information from published patents since the Federal Circuit does not require would-be patentees to disclose their inventions' source code.¹⁰³ Software manufacturers may also try to counteract reverse engineering by patenting a small portion of a program, such as a lock-out device, and enforcing that patent.

In sum, although *Connectix* removes copyright liability for reverse engineering, this decision ironically may diminish reverse engineering by encouraging software companies to patent their software and then sue emulator-makers for patent infringement. It is possible that such patent

97. Software patents present many complicated legal issues. The merits of software patents are beyond the scope of this Note.

98. See *Connectix*, 203 F.3d at 598.

99. See *Sony Continues Battle Against Bleem and Connectix*, CONSUMER MULTIMEDIA REP., May 29, 2000.

100. See *id.*; *Sony Computer Entm't America, Inc. v. Bleem LLC*, 214 F.3d 1022 (9th Cir. 2000).

101. After years of equivocation, the Federal Circuit in *In re Alappat* endorsed patentability of software without qualification. 33 F.3d 1526, 1543 (Fed. Cir. 1994). "Close to one hundred thousand software or software-related patents are now in force in the United States . . ." Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CALIF. L. REV. 1, 11 (2001).

102. "The patent statute includes no express provision allowing reverse engineering, nor is there any judicially-developed exception akin to copyright's fair use doctrine that might permit it." *Id.* at 18.

103. See *id.* at 24 n.87.

infringement claims may be barred by the first sale doctrine¹⁰⁴ or may leave plaintiffs vulnerable to patent misuse counterclaims¹⁰⁵ or antitrust suits.¹⁰⁶ Yet, for now, software manufacturers will increasingly rely on software patents to curb competition from the emulator-makers.

2. *Greater Use of Shrinkwrap Licenses*

Greater use of shrinkwrap licenses in the software industry is another strategy available to software manufacturers seeking to protect their products from reverse engineering.¹⁰⁷ Generally, a shrinkwrap license is a standardized, unsigned agreement between the purchaser of a software program and the software manufacturer.¹⁰⁸ It defines the terms of the transaction and places contractual conditions on the purchaser's use of the software.¹⁰⁹ These shrinkwrap licenses could potentially restrict reverse engineering of software, arguably bypassing copyright law by relying on contract law. The courts have affirmed the validity of shrinkwrap licenses,¹¹⁰ and they are widely used in connection with the sales of computer products.¹¹¹ In the future, these licenses are likely to be governed by

104. See Cohen & Lemley, *supra* note 101, at 30-35. The first sale doctrine states that patentees who have introduced their invention to the marketplace are estopped from claiming patent infringement against anyone obtaining the invention from the original purchaser. Mark J. Rozman, *Intel v. ULSI System Technology, Inc.*, 1 J. INTELL. PROP. L. 373, 379 (1994).

105. See *id.* The patent misuse doctrine allows the courts to deny remedy for patent infringement to patent owners who used their patent in an improper manner, for example, by violating the antitrust laws or extending the patent beyond its lawful scope. Note, *Is the Patent Misuse Doctrine Obsolete?*, 110 HARV. L. REV. 1922, 1923 (1997).

106. "Patent misuse is frequently . . . coextensive with conduct that violates the anti-trust laws." Cohen & Lemley, *supra* note 101, at 35.

107. See Page M. Kaufman, *The Enforceability of State "Shrink-wrap" License Statutes In Light Of Vault Corp. v. Quaid Software, Ltd.*, 74 CORNELL L. REV. 222, 224 (1988) (stating that software developers often attempt to use "shrink-wrap" license agreements to broaden the protection offered under the Copyright Act).

In this section, I am making two assumptions: (1) that software manufacturers either will not be able to use their software patents to prevent reverse engineering or will choose to rely on enforcement of shrinkwrap licenses instead, and (2) that although shrinkwrap licenses are already widely used in connection with software sales, after *Connectix* their use and enforcement will become even more widespread.

108. Darren C. Baker, *ProCD v. Zeidenberg: Commercial Reality, Flexibility in Contract Formation, and Notions of Manifested Assent in the Arena of Shrinkwrap Licenses*, 92 NW. U. L. REV. 379, 389 (1997).

109. *Id.*

110. See, e.g., *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996).

111. Most commercial software is made available pursuant to a shrinkwrap license that typically includes a provision prohibiting the licensee from engaging in any reverse engineering of the software. See Nicholas Groombridge, *Reverse Engineering Copy-*

the Uniform Computer Information Transaction Act ("UCITA"), a model contract law statute.¹¹²

UCITA's purported aim is to promote freedom of contract.¹¹³ By purchasing and opening a product, a consumer can bind himself to an agreement to refrain from reverse engineering any software contained therein. In the mass-market software context, the licensees are devoid of any real bargaining power.¹¹⁴ Agreeing to the provisions of the shrinkwrap license is the unavoidable prerequisite to obtaining a software program. Thus, a consumer may not be able to obtain the software whose functional elements he would like to study without preemptively "agreeing" to the provision prohibiting any kind of reverse engineering whatsoever.

Legal issues raised by shrinkwrap licenses in general, and UCITA in particular, are beyond the scope of this Note.¹¹⁵ It bears noting, however, that *Sony v. Connectix* brings these issues to the forefront more than ever because of the greater reliance on shrinkwrap licenses that is to be expected as result of this decision.

3. Greater Consumer Choice and Improved Product Quality

Sony v. Connectix could have some positive consequences for consumers as well. Consumer choice could be broadened through the creation of new products, including higher-quality emulator software. Legal implications of its actions aside, by creating the VGS, Connectix added to the video game console market by providing video game players with a new platform for the games. Consumers certainly benefit from a wider choice

righted Software?; Ninth Circuit's Recent Ruling Clarifies Law, Resolves Long-Standing Questions, N.Y. L.J., Oct. 16, 2000, at S3.

112. UCITA allows enforceability of shrinkwrap licenses only if three requirements are met. First, the buyer must have reason to expect that additional contract terms will be proposed after the purchase. UNIF. COMPUTER INFO. TRANS. ACT § 208(2), at 119, available at <http://www.law.upenn.edu/bll/ulc/ucita/ucitaFinal00.pdf> (Sep. 29, 2000). Second, the buyer must be able to return the product at the licensor's cost. *Id.* § 209(b), at 122. Finally, the buyer must be able to recover damages for the alterations to his system if it has been altered by the installation of license terms for review. *Id.*

113. *Id.* at 1 (prefatory note) ("UCITA . . . [is] based upon the principle of freedom of contract . . .").

114. See, e.g., Pratik A. Shah, Note, *The Uniform Computer Information Transactions Act*, 15 BERKELEY TECH. L.J. 85, 93 (2000) ("[G]iven the bargaining power of most licensors over licensees in the mass-market shrinkwrap context, where adhesion contracts are the norm, this apparent efficiency could come at the licensee's expense.").

115. See *id.*; see also Brian D. McDonald, Note, *The Uniform Computer Information Transactions Act*, 16 BERKELEY TECH. L.J. 461 (2001) (discussing the prospects for widespread adoption of UCITA).

of video game consoles.¹¹⁶ Given the emulator industry-friendly decision in *Connectix*, other small software companies will be encouraged to exploit larger companies' reluctance to enter new markets by creating and marketing emulator software. Reverse engineering of software is inexpensive, at least relative to its creation. In creating the VGS, Connectix spent less than 0.01% of Sony's research and development and marketing budgets for the PlayStation.¹¹⁷ Small start-up companies also lack larger companies' inherent bureaucratic resistance to quick production and marketing of new emulator products.

Alternatively, to compete with the emulator industry, large software manufacturers might expand their product lines, consequently expanding consumer choice. Instead of preparing a single version of their product, large companies could be driven to create a number of versions compatible with different platforms in order to preempt competing emulator software. Creation of several versions of products by large companies may improve the quality of emulator products. Connectix's VGS was flawed and "buggy,"¹¹⁸ but a company like Sony with its superior resources would be able to create high quality emulators. Also, manufacturers of the original software would not have to engage in the time-consuming and imperfect process of reverse engineering. With full access to both source and object code of the original, original developers can improve the quality of emulator products.

IV. CONCLUSION

Copyright law has to adapt its traditional principles to a continually expanding field of subject matter. The courts faced a particularly difficult challenge when software, a utilitarian article, was granted copyright protection in 1980. *Sony v. Connectix* reinforces the rule that was established by such cases as *Atari v. Nintendo* and *Sega v. Accolade*—reverse engineering of copyrighted code performed for the purpose of accessing the

116. See, e.g., Sam Pettus, *Emulation: Right or Wrong?*, at http://www.emuhq.com/emufaq/mod3_pt1.htm (Oct. 1, 1999) ("[E]mulation actually allows users to enjoy programs outside of their intended platform. Mark Asher, writing for C|NET GameCenter, puts it this way: 'What this denial of injunction [for Sony] means for gamers is simple: freedom of choice.' Howard Wen, writing for Salon, calls it "'tearing down the barriers for code among proprietary formats.' . . .").

117. See *Sony Computer Entm't, Inc. v. Connectix Corp.*, 48 F. Supp. 2d 1212, 1214 (N.D. Cal 1999) (stating that Sony spent over \$600 million developing the PlayStation); see also *Sony Computer Entm't, Inc. v. Connectix Corp.* 203 F.3d 596, 601 (9th Cir. 2000) (stating that Connectix spent approximately \$150,000 developing the VGS).

118. *Connectix* at 596.

work's functional elements is fair use. *Connectix* also helps to clarify the holding of *Sega* by specifying that when the necessity of reverse engineering to get to the ideas contained within software is established, the frequency of copying of protected material is irrelevant to the fair use analysis. The Ninth Circuit established that the functional identity between the original and the new work is immaterial when considering the "effect on the market" factor of the fair use test in reverse engineering cases. This holding allows for creation of software whose functions substantially overlap with those of the work from which it was reverse engineered.

Sony v. Connectix, for the first time, establishes the applicability of the fair use defense to the production of emulator products through reverse engineering. Although the court's analysis of the fourth fair use factor is flawed, the software industry will likely be able to counteract the effect of *Connectix* by enforcing software patents, relying on shrinkwrap licenses that prohibit reverse engineering, and preemptively producing emulator products.