Can Black Dot (Shrinkwrap) Licenses Override Federal Reverse Engineering Rights?: The Relationship Between Copyright, Contract, and Antitrust Laws

Seungwoo Son[*]

I.   Iɴᴛʀᴏᴅᴜᴄᴛɪᴏɴ

    Statutory clarification of reverse engineering rights is necessary to strike a proper balance between providing adequate copyright protection and promoting public access to a copyrighted work. A major fair-access issue has arisen in situations where a copyright holder attempts to restrict a reverse engineering right by copyright infringement litigation, contractual shrink-wrap licenses, or protective system even though the reverse engineering is permissible under the "fair use" doctrine.[1] Reverse engineering of computer software is defined as the process of studying a computer program to obtain useful and detailed insight into the functional mechanisms of the work. Reverse engineering is inevitably involved when making copies of an original computer program to understand the structure and functionality unprotected by copyright; the copyright fair use doctrine may excuse such infringing uses.

    Although Congress has not established clear guidelines, courts have faced this issue and developed a reasonable standard under copyright law to separate legitimate from illegitimate reverse engineering.[2] In addition, courts have ruled that software reverse engineering is generally

---

    1.   *See* 17 U.S.C. § 107 (2000).
    2.   *See, e.g.*, Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000); Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1539 n.18 (11th Cir. 1996); 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).

acceptable for the purpose of obtaining access to unprotectable functional elements if an end product did not encompass any code of the original program. Nevertheless, there has been much controversy concerning the extent to which reverse engineering should be allowed because a software vendor is still able to forbid reverse engineering by using a contractual license (e.g., shrink-wrap licenses or click-wrap licenses) or protective system.

A recently prevalent assumption that "to better promote the industry, greater protection of the industry's products are necessary"[3] has led to most striking legislative actions: the Digital Millennium Copyright Act (DMCA)[4] and the Uniform Computer Information Transactions Act (UCITA).[5] As a result of the current interplay between copyright law and this legislation, uncertainty with respect to reverse engineering has been compounded. These laws undermine the copyright balance by unreasonably narrowing the scope of "fair use" rights and likely bringing about anticompetitive effects in the market. For example, anticircumvention provisions within legislation are not subject to the well-established fair use copyright defense.[6] Thus, even legitimate reverse engineering, other than that performed for interoperability purposes, which circumvents a security measure solely for the purposes of accessing uncopyrighted functional elements would be condemned.

Firms usually attempt to prohibit reverse engineering of their software by employing various contractual devices. As a famous example, most off-the-shelf software purchased in the open market contains a shrink-wrap license. Beneath the clear outer wrapping is a visible license agreement which states that by removing the plastic shrink-wrap enclosing the software, the end user agrees to the terms of the license.[7] This shrink-wrap license almost always includes a prohibition on reverse engineering, including disassembly.[8]

---

3.   Jon M. Garon, *Media & Monopoly in the Information Age: Slowing the Convergence at the Marketplace of Ideas*, 17 CARDOZO ARTS & ENT. L.J. 491, 571 (1999).

4.   Pub. L. No. 105-304, 112 Stat. 2863 (1998).

5.   Draft for Approval (NCCUSL Annual Draft Meeting, July 23-30, 1999), *at* http://www.law.upenn.edu/bll/ulc/ucita/citam99.htm (last visited Dec. 17, 2002); *see also* Proposed Changes to UCITA, http://www.nccusl.org/nccusl/UCITA-2001-comm-fin.htm (last visited Dec. 17, 2002).

6.   107 U.S.C. § 1201 (2000).

7.   *See* Greg Weiner, *Reverse Engineering as a Method of Achieving Compatibility in the Computer Industry*, 6 U. BALT. INTELL. PROP. L.J. 1, 5 (1997).

8.   *See* JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL; INTELLECTUAL PROPERTY AND INTEROPERABILITY IN THE GLOBAL SOFTWARE INDUSTRY 220 (Westview Press 1995).

After the United States Court of Appeals for the Seventh Circuit's decision in *ProCD, Inc. v. Zeidenberg*, shrink-wrap licenses have generally become enforceable.[9] Indeed, UCITA would make the terms of shrink-wrap licenses more enforceable and would outlaw reverse engineering by adopting the *ProCD* court's position.[10] However, UCITA has faced much criticism including the proposed statutes' wording that makes it difficult to understand.[11] Although federal copyright law is not generally intended to preclude private contracts, a contract term may be unenforceable under the purpose of effectuating a compelling federal policy such as the copyright fair use privilege. An antireverse engineering provision under a shrink-wrap license is a case in point. Hence, the National Conference of Commissioners on Uniform State Laws (NCCUSL) has attempted to revise several controversial provisions including the reverse-engineering issue.

However, the debate regarding a shrink-wrap license prohibiting reverse engineering has recently brought about more tension between contract law and intellectual property (IP) laws. The United States Court of Appeals for the Federal Circuit has issued a new opinion in *Bowers v. Baystate Technologies, Inc.*, upholding its original conclusion that a shrink-wrap license can bar reverse engineering of a computer program.[12] Enforcement of a shrink-wrap license term reveals the tension between state contract law and preemption by federal patent, copyright, and trademark law. Professor Mark Lemley, in an amici curiae brief, was concerned that the Panel in the case had gone to the opposite extreme, adopting a blanket rule that such restrictions should never be preempted by federal IP laws.[13] Adopting the same viewpoint, this Article argues that an antireverse engineering term contained in a shrink-wrap license could not only present an obstacle to the operation of the federal IP system, but also discourage the creative activity behind the constitutional

---

    9.   *See* ProCD v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996).
    10.   *See* Draft for Approval (NCCUSL Annual Draft Meeting, July 23-30, 1999), *supra* note 5.
    11.   Proposed Changes to UCITA, *supra* note 5. *See generally* Pamela Samuelson & Kurt Opsahl, *How Tensions Between Intellectual Property Policy and UCITA Are Likely to Be Resolved*, 570 PLI/Pat 741 (1999); Rochelle Cooper Dreyfuss, *UCITA in the International Marketplace: Are We About to Export Bad Innovation Policy?*, 26 BROOK. J. INT'L L. 49 (2000); Jeffrey A. Modisett & Cindy M. Lott, *Cyberlaw and E-Commerce: A State Attorney General's Perspective*, 94 NW. U. L. REV. 643 (2000); Leo L. Clarke, *Performance Risk, Form Contracts and UCITA*, 7 MICH. TELECOMM. & TECH. L. REV. 1 (2000/2001).
    12.   Bowers v. Baystate Techs., Inc., 320 F.3d 1317, 1326 (Fed. Cir. 2003).
    13.   *See* Mark Lemley, Brief Amicus Curiae in Support of Petition for Panel Rehearing in *Bowers v. Baystate Technologies* 9 (Sept. 2002), *available at* http://www.law.berkeley.edu/institutes/bclt/pubs/lemley/bowers.pdf (last visited Feb. 10, 2003).

foundation of IP legislation. Indeed, such a contractual term may be unenforceable because it is so one-sided and unfair as to be unconscionable, which is no different in substance from a black dot law.[14]

　To develop new competitive paradigms which accommodate the rapidly changing technological and business environment, the scope of copyrights needs to be redefined after a substantive discussion of the interaction between contract and antitrust laws with the goal of IP law. Crucial to this argument is that the rights conferred on copyright holders are limited rights. IP laws grant certain exclusive rights or monopoly privileges to inventors or authors, but the privileges are ultimately associated with public interests. Courts and legislatures saw the need to limit the scope of IP monopoly. The copyright "fair use" doctrine reflects this need for limitations on copyrights. The fair use doctrine exempts a reverse engineer from liability incurred by certain decompilation of copyrighted works that would otherwise infringe.[15] The most striking feature of this doctrine is its attempt to limit a copyright owner's exclusive right in order not to stifle creative efforts by competitors through exploring information committed to the public domain through the grant of copyrights.

　This public policy behind the fair use doctrine is important when analyzing reverse engineering issues in conjunction with another characteristic in the high-tech industry: innovation accomplished through a succession of incremental improvements. The resulting mutual dependence of manufacturers yields a "network effect."[16] "Network effect" greatly affects the development of software products if a large number of compatible products—with an industry standard possessed by a dominant market entity—are available in the market. Software products produced by competitors often must have compatibility with an industry standard because they benefit from the network effects of the standard's widespread adoption. It permits competition among providers, which also benefits the market.[17]

14.　In dissenting from the majority opinion in *Bowers*, Judge Dyk held that "[i]f state law provided that a copyright holder could bar fair use of the copyrighted material by placing a black dot on each copy of the work offered for sale, there would be no question but that the state law would be preempted." *Bowers*, 320 F.3d at 1336 (Dyk, J., dissenting).

15.　Decompilation is one of the reverse engineering methods to observe source code in computer software.

16.　*See* Richard N. Langlois, Technological Standards, Innovation, and Essential Facilities: Toward a Schumpeterian Post-Chicago Approach 24 (Dec. 1999) (Paper for the George Mason University conference on Dynamic Competition and Antitrust, December 16-17, 1998, Washington, DC).

17.　*See* Mark A. Lemley & David McGowan, *Legal Implications of Network Economic Effects*, 86 CAL. L. REV. 479, 525 (1998).

Therefore, effective competition requires both computer software and hardware developed by competitors to be compatible with an existing standard system. The reverse engineering right in network industries should be allowed in order to accomplish compatibility. Innovations in the network industry are less of a public good if a large number of distinct participants cannot employ the ideas simultaneously. Thus, defining the scope of access right to the protected work is important.

In general, most efforts by firms to restrict reverse engineering of their software, and corresponding agreements by customers not to use reverse engineering of their supplier's software, are not likely to raise significant antitrust issues.[18] However, it will be a hot-button issue if a software vendor that has market power in the primary market intends to eliminate existing or potential competition by unilaterally imposing antireverse engineering terms, and in so doing brings about anticompetitive effects in the relevant markets. Likewise, if a restraint imposed by the software vendors primarily functions to limit the reverse-engineering rights of a licensee, which is allowed under the fair use doctrine, it will be subject to antitrust scrutiny. Importantly, the software industry represents strong network effects. One convincing argument is that network effects support permitting a reverse engineering right in order to achieve compatibility with a dominant entity since dynamic competition benefits from network effects, which will increase consumer welfare.

The primary goal of this Article is to propose a clarification of the reverse engineering right, which policymakers should draw in terms of copyright, contract, and antitrust laws. My discussion proceeds in three parts. Part II examines some of the key reverse engineering questions raised by recent cases or legislation and suggests what Congress might have done. The idea/expression dichotomy of computer programs, the copyright fair use doctrine, misuse doctrine, and the anticircumvention provisions of DMCA are the subjects crucial to this discussion. Part III discusses the federal copyright law preemption of state contract law because a shrink-wrap license always includes a provision prohibiting any reverse engineering that may be legitimate activity under the Copyright Act. This Part especially criticizes uncertainty around UCITA that provides software vendors with power to enforce their shrink-wrap license even when it contains terms against public policy. Finally, Part

---

    18.   *See* Robert H. Lande & Sturgis M. Sobin, *Reverse Engineering of Computer Software and U.S. Antitrust Law*, 9 HARV. J.L. & TECH. 237, 238 (1996).

IV attempts to articulate the anticompetitive effects of possible antireverse engineering activities in various hypothetical situations where they may be subject to violation of § 1 or § 2 of the Sherman Act. The proposed analysis in this Part supports a proposition that use of competitive criteria in resolving the reverse engineering right does not reduce the opportunity to use copyright as an exclusionary strategy.

II.   NATURE OF SOFTWARE REVERSE ENGINEERING

Reverse engineering is a process for understanding the structure and functionality of software, given the source code.[19]   Thus, reverse engineering is often employed for understanding new innovations by a competitor and achieving compatibility between two products or programs.[20] Goals of reverse engineering include developing competing software or hardware peripherals and compatible new products, uncovering the weaknesses of products, and satisfying idle curiosity.[21]

In the typical software development process, computer programs are initially written in an alphanumeric language that consists of words and arithmetical expressions meaningful to humans (source code).[22] The source code is then translated or compiled by a utility program into a computer-readable code (object code).[23] Reverse engineering involves obtaining either the original source code or detailed written specifications from the original programmer. Otherwise, it must be undertaken independently by decompiling the object code back into the source code. As a practical matter, it may be impossible to reverse engineer a computer program without decompiling the object code back into an equivalent source code version.[24] With respect to fair use arguments in copyright law, it is significant to understand that the process

---

        19.   The concept of reverse engineering is employed in almost every field of inventions including software and hardware. This Article focuses on a controversial reverse engineering issue surrounding computer programs.
        20.   *See generally* E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1490 (D. Minn. 1985).
        21.   *See* David Bender & M. Elaine Johnston, *Antitrust Aspects of Reverse Engineering*, 365 PLI/IP & ANTITRUST 709, 715 (1993).
        22.   For example, such source programming languages are BASIC, C+, COBOL, FORTRAN, PASCAL, PL/I, ADA, etc.
        23.   The object code is output from a compiler or assembler which is itself an executable machine code or is suitable for processing to produce executable machine codes. The object code simply consists of a series of "0's" and "1's."
        24.   1 ROGER M. MILGRIM, MILGRIM ON TRADE SECRETS § 1.05[5], at 234 (2002); Lande & Sobin, *supra* note 18, at 240-41 ("Because of factors inherent in the present technology, it is practically impossible to decompile object code back into an exact replica of the original source code.").

of decompilation generally cannot be achieved without at some point making copies of the original program.[25]

    Reverse engineering is not inherently unlawful: the process has been occurring for many decades within the industrial and scientific communities.[26] Copyright, trade secret, and even patent laws implicitly endorse the legality of reverse engineering.[27] Its legality under copyright law depends on the particular purpose of reverse engineering. The United States Supreme Court has defined 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."[28]

    In this respect, the motivations and contexts of software reverse engineering are divided into two major categories: developing noncompeting vs. competing programs. Within this distinction, there are many subdistinctions including making copies for using part or all protected features of the original program,[29] error correction, improvement, functional equivalents,[30] and interoperability.[31] Making a

---

    25.   Lande & Sobin, *supra* note 18, at 241 (copying may take the forms of loading the program into computer memory, outputting it to a screen or printer, or copying it to other media).

    26.   *See* Craig L. Uhrich, *Economic Espionage Act-Reverse Engineering and the Intellectual Property Public Policy*, 7 MICH. TELECOMM. & TECH. L. REV. 147, 157 (2000-2001); Robert G. Bone, *A New Look at Trade Secret Law: Doctrine in Search of Justification*, 86 CAL. L. REV. 241, 291 n.219 (1998) (describing how the semiconductor industry lobbied Congress to allow reverse engineering as a step toward creating an improved chip); Bender & Johnston, *supra* note 21, at 709.

    27.   Patent law does not directly address the legitimacy of reverse engineering, but the legality might depend on whether it involves the making or using of patented subject matter. The "experimental use" defense may be raised in patent infringement claims based on reverse engineering. For a detailed discussion of copyright law, see Part III; UCITA § 105 cmt. 3 (2002) ("[T]rade secret law does not prohibit reverse engineering of lawfully acquired goods available on the open market.").

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

    29.   There is a substantial similarity or identification between the protected work and the final work created by using information regarding the arbitrary structure, functionality, ideas, and expression embodied in the original program. In *Bowers v. Baystate Technologies, Inc.*, the court found extensive and unusual similarities between Geodraft and the accused Draft-Pak in structure and design, including idiosyncratic design choices and inadvertent design flaws. 320 F.3d 1317, 1326 (Fed. Cir. 2003).

    30.   There is a desire to produce a new program that is functionally equivalent to the original work. The distinction between this type and the direct copy is that the developer of the new code aims to enable his product to perform the same function as the code that has been studied, but achieves that functionality by code structure and procedures developed independently (not copied) from the original code. *See* Lande & Sobin, *supra* note 18, at 241-42. In this context, reverse engineering is used in a "clean room" technical process, by which one team of programmers decompile the program or chip to isolate the unprotected, noncopyrightable, functional components. Then, the engineers give those functional components to another team who never had access to the original code. The clean room technique is preferred because it insulates the second team of engineers from the original source code. The second team develops

copy is a prima facie case of infringement because the initial result of the process of reverse engineering is substantially similar to the protected work. When it comes to the purposes of functional equivalents and interoperable products, however, the fair use defense for intermediate copying occurring during decompilation may be considered. Through copyright and patent legislation, Congress and the courts have recognized the proprietary nature of software programs.[32]

Historically, most computer programs were designed to perform independently from other programs making interoperability less important than it is today. However, technology advances the value of a computer program to its users while also making it increasingly dependent on compatibility or interoperability with a particular hardware or software system.[33] Understanding a protected computer system or operating programs by using the technology of reverse engineering is obviously necessary for new application development and competition. In reality, however, a copyright owner of an operating system or software application expends great efforts to prevent reverse engineering of its program by its competitors or third parties by way of using copyright and patent protections and creating technical tools such as a password or "lock-and-key" programs.[34] Such situations give rise to the most contentious legal question in recent decades: how the "fair use doctrine" resolves the underlying intermediate copying[35] during the decompilation of a copyrighted program for the purpose of compatibility with the

---

their own code to achieve the desired functionality based on only the necessary information of the original program for functional equivalence. *See* Christopher W. Hager, *Apples and Oranges: Reverse Engineering as a Fair Use After* Atari v. Nintendo *and* Sega v. Accolade, 20 RUTGERS COMPUTER & TECH. L.J. 259, 265 (1994).

    31.   Interoperability is defined as "the ability of computer programs to exchange information, and for such programs mutually to use the information which has been exchanged." S. REP. NO. 105-190, at 33 (1998). The value of interoperability is great when the system or products conform to a standard that is widely accepted in the market. Such effects are called "network effects."

    32.   *See* NATIONAL COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS (1979).

    33.   *See* Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 S. CAL. L. REV. 1091, 1093 (1995); Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 JURIMETRICS J. 33, 63-64 (1987); Timothy S. Teter, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 STAN. L. REV. 1061, 1063-66 (1993).

    34.   *See* Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 256 (5th Cir. 1988).

    35.   Intermediate copies of a software program include "the computer file generated by the disassembly program, the printouts of the disassembled code, and the computer files containing . . . modifications of the code that were generated during the reverse engineering process." Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1518 (9th Cir. 1992).

program. Furthermore, under what circumstances should decompilation be allowed as a legitimate use within copyright protection?

   There are roughly four methods of observation and analysis under the umbrella of software reverse engineering: (1) read about the program in the manuals, (2) observe the program in operation by loading and running it on a computer and viewing what the program does on the screen displays, (3) perform a static examination of the individual computer instructions contained within the program, or (4) perform a dynamic examination of instructions as the program is being run on a computer.[36]

   The first method of software reverse engineering is the least effective method and requires gathering and reading relevant documentation; e.g., a "Readme" file or the manuals concerning the computer program. However, manufacturers may not be aware of all the situations in which their software will not work properly until it has been shipped to buyers. This problem may cause software documentation to be outdated or incomplete.[37] Hence, when a reverse engineer intends to diagnose an unexpected failure of computer software, reading manuals will rarely provide the requisite information.[38]

   The second method involves copying the copyrighted program into the computer's random access memory or RAM, but does not require disassembly of the object code of a protected work.[39] This is accomplished differently, depending on the type of the software program. The ideas and functional elements of some software programs, e.g., word processing programs, spreadsheets, and video game displays, are readily discernible by observation of the external visual expression of the object code's operation on the computer screen.[40]

   The observations for other programs, such as operating systems, are more complex because the functional elements of these programs are not visible to the user during their operation.[41] To analyze these programs, methods have been developed to reproduce the program in an emulated

---

   36.   *See* Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 600 (9th Cir. 2000); Andrew Johnson-Laird, *Software Reverse Engineering in the Real World*, 19 U. DAYTON L. REV. 843, 846 (1994).
   37.   *See* Johnson-Laird, *supra* note 36, at 846.
   38.   *See id.*
   39.   *See Sony Computer Entm't*, 203 F.3d at 600 (describing how § 117(a)(1) provides for copying of copyrighted software programs into RAM shall not be an infringement because a buyer must copy the program into the memory of a computer in order to make any use of the program).
   40.   *See id.*; *Sega*, 977 F.2d at 1520.
   41.   *See* Teter, *supra* note 33, at 1063-66.

environment[42] while simultaneously running research tools such as a debugger.[43] During the execution of this method, the computer program is directly copied each time the engineer turns on the computer.[44] However, all of this copying may be intermediate, which means none of the protected material may be copied into, or appear in, the end product.[45]

The third and fourth methods of reverse engineering are pertinent to decompilation of the object code back into source code.[46] In each case, "engineers use a program known as a 'disassembler' [or decompiler] to translate the ones and zeros of binary machine-readable object code into the words and mathematical symbols of source code."[47] In a static examination of the computer instructions, an engineer decompiles the object code of all or part of the program without operating the program itself.[48] In the fourth method of reverse engineering, a dynamic examination of the computer instructions, an engineer uses a decompiler to decompile parts of the program, one instruction at a time, while the program is running.[49] Depending on the number of times this operation is performed, this method requires copying the program. Both methods of reverse engineering involve direct copies of the copyrighted software, but when such instances occur, they are intermediate.

---

42. "Emulation" refers to the ability of a program or device to imitate another program or device. An "emulator" is a hardware or software device that pretends to be another particular device or program that other components expect to interact with. *See* Search390 Definition, *available at* http://search390.techtarget.com/sDefinition/0,,sid10_gci214357,00.html (last visited May 16, 2002). For example, consider a printer that emulates a Hewlett Packard printer so that a computer can communicate with it through a Hewlett Packard printer driver, or it can also work with any software written for an HP printer; or a Macintosh that emulates a PC so it can run PC programs. *See* Computeruser Dictionary, *available at* http://www.computeruser.com (last visited May 16, 2002). In the *Sony* case, Connectix engineers emulated Sony's PlayStation's system to develop its own software program that allows PlayStation games to be played on PCs. *See Sony*, 203 F.3d at 598.

43. The term "debugger" comes from the term "bug," defined as a mistake that a user wants to eliminate. A debugger is a programmer that actually fixes bugs. The debugger's job is to help expose such bugs. Debuggers are visibility tools that make a program visible to the user in a variety of ways. Thus, the debugger is also used beyond the scope of finding errors. *See* Jeffrey Lawrence Korn, Abstraction and Visualization in Graphic Debuggers 2-3 (1999) (unpublished Ph.D. dissertation, Princeton University) (on file with ProQuest Digital Dissertations). It allows the computer software engineer to trace through the application program and observe how the program or part of a correctly functioning program, works when run in conjunction with the application program to be analyzed. *See* PETER NORTON & JOHN SOCHA, PETER NORTON'S ASSEMBLY LANGUAGE BOOK FOR THE IBM PC 5 (1986).

44. *Sony*, 203 F.3d at 600 ("[T]he software program is copied each time the engineer boots up the computer, and the computer copies the program into RAM.").

45. *See id.*

46. *See id.*

47. *Id.*

48. In this process, the program must generally be copied one or more times. *See id.*

49. *See id.*

III.   SCOPE OF COPYRIGHT PROTECTION FOR SOFTWARE REVERSE
       ENGINEERING

A.   Idea vs. Expression Dichotomy of Computer Programs

1.   Protectable Expression and Unprotectable Idea

   Copyright protection does not extend to an idea itself, only the expression.[50]   Section 102(b) of the Copyright Act states: "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."[51]

   The idea/expression dichotomy serves the primary purpose of copyright law, to promote creativity and disseminate useful arts, so that the public may benefit from the labor of authors.[52]   It provides a balance between the need to reward artists and the need for free access to ideas, but often gives rise to a tension between copyright and free speech interests.[53]   First Amendment considerations define the idea side of the copyright dichotomy, which must be kept as a public preserve.[54]

   In *Eichel v. Marcin*, the court held that

> [i]f an author, by originating a new arrangement and form of expression of
> certain ideas or conceptions, could withdraw these ideas or conceptions
> from the stock of materials to be used by other authors, each copyright

---

      50.   The case law has developed the doctrine of "the idea/expression dichotomy."   *See, e.g.*, Baker v. Selden, 101 U.S. 99, 104 (1879) (holding that copyright protects author's explanation of bookkeeping method, but not method itself); Dymow v. Bolton, 11 F.2d 690, 691 (2d Cir. 1926); Nichols v. Universal Pictures Corp., 45 F.2d 119, 122 (2d Cir. 1930) (holding that copyright on play does not extend to basic idea of lovers of different backgrounds and parental disapproval of their relationship), *cert. denied*, 282 U.S. 902 (1931); Universal Pictures Co. v. Harold Lloyd Corp., 162 F.2d 354, 363 (9th Cir. 1947); Mazer v. Stein, 347 U.S. 201, 217 (1954) (holding that copyright extends to particular form of plaintiff's statuettes, but not to idea of using statuette of human figures as lamp base); Reyher v. Children's Television Workshop, 533 F.2d 87, 90 (2d Cir.), *cert. denied*, 429 U.S. 980 (1976);   Sid & Marty Krofft Television Prods. v. McDonald's Corp., 562 F.2d 1157, 1168 (9th Cir. 1977) (postulating that Michelangelo's "David" would be entitled protection as expression while idea of statute of nude male does not); Warner Bros. v. Am. Broad. Cos., 720 F.2d 231, 239 (2d Cir. 1983).
      51.   17 U.S.C. § 102(b) (2000).
      52.   Leslie A. Kurtz, *Speaking to the Ghost: Idea and Expression in Copyright*, 47 U. MIAMI L. REV. 1221, 1224 (1993).
      53.   Triangle Publ'ns, Inc. v. Knight-Ridder Newspapers, Inc., 626 F.2d 1171, 1178 (5th Cir. 1980) (Brown , J., concurring); Neil Weinstock Netanel, *Locating Copyright Within the First Amendment Skein*, 54 STAN. L. REV. 1, 8 (2001); Melville Nimmer, *Does Copyright Abridge the First Amendment Guarantees of Free Speech and Press?*, 17 U.C.L.A. L. REV. 1180 (1970).
      54.   Justin Hughes, *The Philosophy of Intellectual Property*, 77 GEO. L.J. 287, 314 (1988) ("Labor defines the 'expression' side—that which must be rewarded because it is unpleasant activity.").

would narrow the field of thought open for development and exploitation, and science, poetry, narrative, and dramatic fiction and other branches of literature would be hindered by copyright, instead of being promoted.[55]

In terms of economic efficiency, ideas or concepts, because they are the basic building blocks of creation, should be free for use without having to locate and bargain with earlier authors.[56] The dichotomy of idea/expression is marginal or negative where the protection is given to an author's idea.

The two fundamental prerequisites for copyright protection under the 1976 Copyright Act are (1) originality and (2) fixation in a tangible form.[57] Unlike patent law, this does not require novelty or ingenuity,[58] but the originality must be independently created by the author and satisfy a minimal level of creativity.[59] A work is regarded as fixed in a tangible medium of expression when "its embodiment [is] in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."[60]

The history of copyright law gradually expanded in the categories of protected works. Scientific discoveries and technological developments make possible new forms of creative expression that never existed before. For example, electronic music, filmstrips, and computer programs are considered an extension of copyrightable subject matter.[61] Written computer programs are also protected by copyright law as literary works.[62] A computer program, so long as it is original and sufficiently expressive, will be copyrightable whether it is in source code

---

55.   Eichel v. Marcin, 241 F. 404, 408 (S.D.N.Y. 1913); *see also* Int'l News Serv. v. Associated Press, 248 U.S. 215, 250 (1918). Justice Brandeis said, "the noblest of human productions—knowledge, truths ascertained, conceptions, and ideas—become, after voluntary communication to others, free as the air to common use." *Id.*(Brandeis, J., dissenting).

56.   *See* William M. Landes & Richard A. Posner, *An Economic Analysis of Copyright Law*, 18 J. LEGAL STUD. 325, 347-53 (1989); Kurtz, *supra* note 52, at 1224-25.

57.   17 U.S.C. § 102(a) (2000).

58.   *See* Baker v. Selden, 101 U.S. 99, 102 (1880).

59.   *See* Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 345 (1991).

60.   17 U.S.C. § 101 (2000).

61.   *See* PAUL GOLDSTEIN, COPYRIGHT, PATENT, TRADEMARK AND RELATED STATE DOCTRINES § 3.III.2, at 582 (1999).

62.   *See* Pub. L. No. 94-553, 90 Stat. 2541 (1976) (codified as amended at 17 U.S.C. §§ 101, 102, 117 (1988 & Supp. 1994)); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 855-56 (2d Cir. 1982); Williams Elecs., Inc. v. Arctic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982). Computer programs can also be protected by patent law. *See* Diamond v. Diehr, 450 U.S. 175 (1981).

or object code, and whether it is embodied in paper, computer chip, magnetic disk, tape.[63]

     As for the idea/expression dichotomy of computer programs, the expression adopted by the programmer is the copyrightable subject in a computer program. However, the actual processes, structure, or the method of operation embodied in the program are not within the scope of the copyright protection. Decompiling object code of a computer program using reverse engineering in order to understand the ideas and processes of the program is generally excused in an infringement claim for an author's copyright, while ideas are regarded as in the public domain. If a copyright owner is allowed to prohibit any reverse engineering for the ideas contained in the software, it might increase the cost of creating works and reduce the number of works created.

2.   Abstraction-Filtration-Comparison Analysis for Computer
       Programs

     The United States Court of Appeals for the Second Circuit, in *Computer Associates International, Inc. v. Altai, Inc.*, uses the "abstraction/filtration/comparison" analysis to help distinguish the nonprotected ideas in a computer program from the program's protected expression in cases involving copyright infringement.[64]

     Prior to the *Altai* decision, there was no clear guideline for identifying where the line lies between idea and expression for computer programs because of their unique hybrid nature.[65] In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, the United States Court of Appeals for the Third Circuit attempted to tackle the problem by suggesting that the purpose or function of a computer program was the idea of the program as a whole, and everything that was not necessary to that purpose or function would be part of the expression of that idea.[66] However, the *Altai* court and other courts criticized this approach because it assumed that there was only one idea embodied in a computer program and that once a separable idea could be identified, everything else must be expression.[67]

---

     63.   *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).
     64.   982 F.2d 693 (2d Cir. 1992).
     65.   *See id.* at 703 ("The essentially utilitarian nature of a computer program further complicates the task of distilling its idea from its expression." (citations omitted)).
     66.   797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).
     67.   *Altai*, 982 F.2d at 705 (suggesting that the standard *Whelan* advised was conceptually overbroad); *see also* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1524-25 (9th Cir. 1992). Professor Nimmer pointed out that "[t]he crucial flaw in [Whelan's] reasoning is that it assumes

    In fact, a computer program is executed through the interaction of subroutines, consisting of sequences of codes that are intended to be called and used repeatedly during the operation of the program. Each subroutine is itself a program and has its own idea.[68] Thus, the ultimate purpose or function of a computer program is the composition of each idea contained in the subroutines and sub-subroutines, which should be identified for the test of idea/expression distinction.

    In the abstraction-filtration-comparison analysis, the court first broke down the allegedly infringed program into conceptually separate modules.[69] By examining each of these parts for such things as incorporated ideas, expressions that are necessarily incidental to those ideas, and elements that are taken from the public domain, the court would enable another court to sift out all nonprotectable elements. The court applied the doctrine of merger[70] and the "scenes à faire" doctrine[71] to computer software at this filtration stage. It held that the merger doctrine precluded copyright protection for the particular set of modules, which was necessary to promote program efficiency or speed, in such instances where its idea could not be separated from the creative and technical expression.[72]

    As the final step, the court isolated the golden nugget protectable code and compared this with the software application at issue. The result of this comparison determined whether the protectable elements of the

---

that only one 'idea,' in copyright law terms, underlies any computer program, and that once a separable idea can be identified, everything else must be expression." 4 M. NIMMER & D. NIMMER, NIMMER ON COPYRIGHT §§ 13.03[F][1], 13-122 (2002).

    68.   *Sega*, 977 F.2d at 1525 ("For example . . . a video game program . . . contains at least two such subroutines—the subroutine that allows the user to interact with the video game and the subroutine that allows the game cartridge to interact with the console.").

    69.   *See Altai*, 982 F.2d at 706-07.

    70.   The principle of the merger doctrine is that "[w]hen there is essentially only one way to express an idea, the idea and its expression are inseparable and copyright is no bar to copying that expression." Concrete Mach. Co. v. Classic Lawn Ornaments, Inc., 843 F.2d 600, 606 (1st Cir. 1988).

    71.   "*Scenes à faire*" are "incidents, characters or settings which are as a practical matter indispensable, or at least standard, in the treatment of a given topic." Alexander v. Haley, 460 F. Supp. 40, 45 (S.D.N.Y. 1978). Therefore, such *scenes à faire* are not copyrightable because "it is virtually impossible to write about a particular historical era or fictional theme without employing certain 'stock' or standard literary devices." Hoehling v. Universal City Studios, Inc., 618 F.2d 972, 979 (2d Cir.), *cert. denied*, 449 U.S. 841 (1980); *see also* Reyher v. Children's Television Workshop, 533 F.2d 87, 91 (2d Cir.), *cert. denied*, 429 U.S. 980 (1976).

    72.   *See Altai*, 982 F.2d at 707-11; *see also* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) (holding that an idea and expression in the computer program merge if there are no or few other ways of expressing a particular idea).

programs were substantially similar so as to warrant a finding of infringement.[73] This better approach has been approved by other courts.[74]

B.   The Fair Use Defense for Software Reverse Engineering

1.   Fair Use Doctrine

The Copyright Act may excuse a reverse engineer from copyright infringement liability if she or he is engaged in a fair use of copyright protected works. The fair use doctrine allows others to use a copyrighted work without the owner's consent in a reasonable manner for certain purposes, such as criticism, comment, news reporting, teaching, scholarship, or research.[75] Copyright law confers upon an author certain exclusive rights.[76] However, some limitations exist on the author's exclusive right to the copyrighted works, since the public benefits from wide dissemination of creative works.[77] Among these limitations, the most important is known as "fair use." An intermediate copy which is made during the process of reverse engineering in order to understand the ideas and functional elements embodied in the copyrighted computer program may be considered to have a "research" purpose as a fair use exception.[78]

---

    73.   *See Altai*, 982 F.2d at 710-12.
    74.   *See, e.g.*, Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1525 (9th Cir. 1992); Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 839 (Fed. Cir. 1992); Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp. 616, 623 (N.D. Cal. 1993).
    75.   17 U.S.C. § 107 (2000); *see also* Infinity Broad. Corp. v. Kirkwood, 150 F.3d 104, 107 n.1 (2d Cir. 1998); Nihon Keizai Shimbun, Inc. v. Comline Bus. Data, Inc., No. 98 Civ. 641 DLC, 1998 U.S. Dist. LEXIS 6806, at *36 (S.D.N.Y. Apr. 14, 1998).
    76.   Section 106 of the Copyright Act grants the owner of a copyright the exclusive right to do and to authorize others to do 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 by sale or other transfer of ownership, or by rental, lease, or lending;
    (4)   in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works, to perform the copyrighted work publicly;
    (5)   in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work, 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.
    77.   *Id.* §§ 107-120.
    78.   *See* PAUL GOLDSTEIN, COPYRIGHT: PRINCIPLES, LAW AND PRACTICE § 5.2.1.4, at 89 (Supp. 1991). The preamble of § 107 lists several uses such as criticism, comment, news

Section 107 of the Copyright Act lists four enumerated factors to be considered in determining whether a particular use constitutes a fair use. These factors include:

(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.[79]

"Fair use is a mixed question of law and fact."[80] The above factors are not exclusive, but rather the fair use doctrine is essentially "an equitable rule of reason."[81] Thus, no generally applicable definition is possible, and the court must be free to adopt the doctrine to particular situations on a case-by-case basis.[82] The doctrine of fair use was initially developed by the English law and equity courts in the late eighteenth and early nineteenth centuries. Although the first Western copyright legislation, the Statute of Anne in 1710, gave no clear guidance to determine infringement, the purpose of the statute is to encourage authors to "Compose and Write Useful Books," to protect the public domain and to provide for public access.[83] Thus, the fundamental justification for fair use defense lies in the purpose of copyright law: "[t]o promote the Progress of Science and useful Arts."[84]

Because fair use is an equitable rule of reason, the court should consider all four statutory factors, and, if necessary, other plausible factors. Other equitable considerations include privacy interests,[85] the

---

reporting, teaching, scholarship, or research, which are presumptively deemed a fair use of copyrighted material. 17 U.S.C. § 107. This preamble paragraph is to indicate an "illustrative and not limitative function" which provides only "general guidance about the sorts of copying that courts and Congress most commonly had found to be fair uses." Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 577-78 (1994); *see also* Harper & Row Publishers, Inc. v. Nation Enters., Inc., 471 U.S. 539, 561 (1985).

    79.   17 U.S.C. § 107.
    80.   *Harper & Row*, 471 U.S. at 560.
    81.   H.R. REP. NO. 1476, 94th Cong., 2d Sess. 65 (1976); S. REP. NO. 473, 94th Cong., 1st Sess. 62 (1975).
    82.   *See Harper & Row*, 471 U.S. at 560.
    83.   *See* 1 WILLIAM F. PATRY, COPYRIGHT LAW AND PRACTICE 718 (1994); *see also* Ray L. Patterson, *The DMCA: A Modern Version of the Licensing Act of 1662*, 10 J. INTELL. PROP. L. 33, 37 (2002).
    84.   U.S. CONST. art. 1, § 8, cl. 8.
    85.   PATRY, *supra* note 83, at 723-24.

age of the copyrighted work,[86] public benefits,[87] a defendant's good-faith[88] and a copyright owner's unclean hands or misuse of the exclusive right to suppress others' publication.[89]

    As will be discussed in Part III.B.4, the application of the fair use doctrine in the context of software reverse engineering is very controversial. Policy arguments supporting denial of access would likely focus on "some of the peculiarities of software markets, such as the importance of protecting large up-front investments in development, the short product lifecycles, and the ease of copying software."[90] Not only did Congress expressly deny copyright protection on the idea, process, or functional concept so as to encourage creative activities, but case law also establishes well the flexibility in the interpretation of the fair use doctrine and even copyright misuse doctrine as an affirmative defense.[91]

    The legislative history of the Copyright Act suggests the flexible application of fair use when courts encounter the problem of fitting new innovations into ancient doctrine. The doctrine of fair use permits courts to avoid an inflexible application of copyright law when under certain circumstances, it would stifle the creative activity which the copyright

---

    86.   *See* Joseph P. Liu, *Copyright and Time: A Proposal*, 101 MICH. L. REV. 409, 410 (2002) ("[T]he older a copyrighted work is, the greater the scope of fair use should be—that is, the greater the ability of others to re-use, critique, transform, and adapt the copyrighted work without permission of the copyright owner.").

    87.   *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1523 (9th Cir. 1992) ("[W]e are free to consider the public benefit resulting from a particular use notwithstanding the fact that the alleged infringer may gain commercially."); discussion *infra* Part III.B.4.f.

    88.   In analyzing the first factor of fair use, courts also consider whether the alleged infringer's conduct is proper. The inquiry turns largely upon whether the defendant acted in bad faith when it copied the copyrighted material. *See* Rubin v. Brooks/Cole Publ'g Co., 836 F. Supp. 909, 918-19 (D. Mass. 1993); Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 562 (1985) ("Fair use presupposes 'good faith' and 'fair dealing.'"); Pierre Leval, *Toward a Fair Use Standard*, 103 HARV. L. REV. 1105, 1126-27 (1990) (arguing that good faith is irrelevant to fair use analysis); Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 585 n.18 (1993) ("[B]eing denied permission to use a work does not weigh against a finding of fair use."); Rogers v. Koons, 960 F.2d 301, 390 (2d Cir.), *cert. denied*, 506 U.S. 934 (1992).

    89.   *See* Rosemont Enters. v. Random House, Inc., 366 F.2d 303, 313 (2d Cir. 1966) ("Rosemont Enterprises acquired the Look copyright and sued upon it six days later asking injunctive relief, not with a desire to protect the value of the original writing but to suppress the Random House biography because Hughes wished to prevent its publication."), *cert. denied*, 364 U.S. 1009 (1967).

    90.   Lande & Sobin, *supra* note 18, at 268.

    91.   H.R. REP. NO. 98-781, 98th Cong., 2 Sess. 23 (1984); Ass'n of Am. Med. Colls. v. Mikaelian, 571 F. Supp. 144, 151 (E.D. Pa. 1983). *See generally* WILLIAM F. PATRY, LATMAN'S THE COPYRIGHT LAW 239-55 (6th ed. 1986). As an affirmative defense, fair use is a privilege, not a right in others than the owner of a copyright to use the protected work in a reasonable manner without the author's consent.

law is designed to foster.[92]   Some commentators have viewed the flexibility of the doctrine as the "safety valve" of copyright law, especially during a period of rapid technological change.[93]   Others have considered the uncertainties of the fair use doctrine the source of unresolved ambiguities.[94]   The Supreme Court in *Sony Corp. v. Universal City Studios, Inc.* accommodated rapid technological change by eschewing a rigid, bright-line approach to fair use.[95]

   The Council Directive of May 14, 1991 on the legal protection of computer programs adopted by the Council of the European Communities allows decompilation of a computer program under certain conditions and when information gathered from the reverse engineering process is to be used to achieve interoperability.   Article 6(1) of the Council Directive states that "[t]he authorization of the rightholder shall not be required where reproduction of the code and translation of its form . . . are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs."[96]

   I conclude that the idea or functional principle contained in computer programs must be subject to the doctrine of fair use because the idea, process, and functional concept should be left free for others to use as basic building blocks.[97]   If decompilation of copyrighted object code is an unfair use per se, it results in a de facto monopoly by the owner of the copyright over the functional aspects of the works.[98]   As for the significant question of under what circumstances reverse engineering for computer programs should be permissible or impermissible, I will analyze more specifically the four underlying factors and other potential factors through recent important legal decisions.

---

   92.   *See* Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 550 (1985); Iowa State Univ. Research Found., Inc. v. Am. Broad. Co., 621 F.2d 57, 60 (2d Cir. 1980).

   93.   *E.g.*, Cassandra Imfeld, *Playing Fair with Fair Use?   The Digital Millennium Copyright Act's Impact on Encryption Researches and Academicians*, 8 COMM. L. & POL'Y 111, 135 (2003); Harry Mihet, *Universal City Studios, Inc. v. Corley: The Constitutional Underpinnings of Fair Use Remain an Open Question*, 2002 DUKE L. & TECH. REV. 3, 11 (2002); *The Supreme Court, 1993 Term Leading Cases II. Federal Statutes and Regulations C. Copyright Law,* 108 HARV. L. REV. 331, 336 (1994).

   94.   *See* H.R. REP. NO. 94-1476, 94th Cong., 2d Sess. 66 (1976); S. REP. NO. 94-473, 9th Cong., 1st Sess. 62 (1975).

   95.   464 U.S. 417, 421 (1984) (holding that the defendant engaged in fair use, thereby allowing customary practices in the video recording industry.)

   96.   Council Directive 91/250/EEC, 1991 O.J. (L. 122), 42-46.

   97.   *See* Douglas L. Rogers, *Give the Smaller Players a Chance: Shaping the Digital Economy Through Antitrust and Copyright Law*, 5 MARQ. INTELL. PROP. L. REV. 13, 77 (2001).

   98.   *See* DSC Communications Corp. v. DGI Techs., Inc., 898 F. Supp. 1183, 1191 (N.D. Tex. 1995), *aff'd in part and rev'd in part*, 166 F.3d 772 (5th Cir. 1999).

2.   *Sega v. Accolade*

In *Sega Enterprises Ltd. v. Accolade, Inc. (Sega)*, the United States Court of Appeals for the Ninth Circuit faced the difficult question of whether the fair use doctrine should be applied to the case of disassembly of computer software in order to achieve compatibility with a copyrighted work.[99]   Sega Enterprises Ltd., the plaintiff, and its subsidiary, Sega of America (collectively referred to as Sega), developed and marketed video game systems, including the "Genesis" console and its corresponding video game cartridges.[100] Accolade, Inc., the defendant, is an independent developer of computer entertainment software, and desired to make video game cartridges that would be compatible with the Genesis game system.[101]

Sega licensed its copyrighted computer code and its "SEGA" trademark to a number of independent developers of computer game software.[102] Those licensees manufactured and sold Genesis-compatible games in competition with Sega's game cartridges.[103] Accolade sought a licensing agreement with Sega, but since the agreement would have required that Sega be the exclusive manufacturer of all games produced by Accolade, it chose not to be a Sega licensee.[104]   Instead, Accolade "reverse engineered" the microcode contained in Sega video game cartridges by using a process of decompiling the binary object code in order to discover the method of interoperating with the Genesis console.[105]

To reverse engineer the Sega games, Accolade employed a two-step process to render its video games compatible with the Genesis console.[106] First, it purchased a Genesis console and three Sega game cartridges that were commercially available.[107] After loading the games into the console, Accolade engineers wired a decompiler into the console and transformed the machine-readable object code contained in the games into human-readable source code.[108] Then, the decompiler generated printouts of the

---

99.   977 F.2d 1510, 1510 (9th Cir. 1992).

100.   *Id.* at 1514.

101.   *See id.*

102.   *See id.*

103.   *See id.*

104.   *See id.*

105.   *See id.* at 1514-15. For more details concerning the process of programming and software reverse engineering, see *supra* Part II.

106.   *See Sega*, 977 F.2d at 1514.

107.   *See id.* at 1514-15.

108.   *See id.* at 1515.

resulting source code.[109] Accolade engineers studied and annotated the printouts to identify common sections of code among the three game programs.[110] They loaded the disassembled code into a computer and experimented in order to understand the interface specifications for the Genesis console by modifying the program.[111] Then, they created an interface specification manual that contained descriptions of only functional interface requirements, but which did not include any of Sega's code.[112]

In the second step, Accolade produced Genesis compatible games based not on copying any of Sega's program, but on the Genesis compatibility development manual.[113] In 1990, Accolade released a video game called "Ishido" for use with the Genesis console.[114]

In 1991, Accolade learned that Sega had developed a new video game console, the Genesis III, which contained a Trademark Security System (TMSS) that prevented piracy of Sega video games.[115] When a game cartridge was inserted, the microprocessor in the Genesis III searched the code for a specific sequence of characters, "S-E-G-A" (the "TMSS initialization code").[116] After locating such a sequence, a display would appear on the screen reading "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD" (the "Sega Message").[117] If the console did not find the sequence at the necessary location in the game program, it would not allow the game to operate.[118]

Further studying of the Genesis III console revealed a small segment of initialization code, approximately twenty bytes of data which turned out to be necessary to bypass the TMSS barrier.[119] Accolade added the code to its development manual in the form of a standard header file to be used in all games.[120] It argued that the code sequence, called a header file, was the only Sega code copied into its products.[121]

---

109. *See id.*
110. *See id.*
111. *See id.*
112. *See id.*
113. *See id.*
114. *See id.*
115. *Id.* At a consumer electronics show, Sega demonstrated that Accolade's "Ishido" game cartridges would not work on the new system. *See id.*
116. *See id.*
117. *See id.*
118. *See id.*
119. *See id.*
120. *See id.*
121. *See id.* at 1516. While each of Accolade's games contained a total of 500,000 to 1,500,000 bytes, the header file contained approximately twenty to twenty-five bytes of data. *See id.*

Sega filed suit against Accolade, alleging that Accolade's intermediate copying of Sega's software during the decompilation portion of the reverse engineering process constituted copyright infringement and the display of the TMSS message constituted trademark infringement.[122] The district court granted Sega's motion for a preliminary injunction against Accolade, enjoining it from selling its reverse engineered games.[123]

Accolade appealed to the Ninth Circuit, which reversed the district court.[124]   The court of appeals agreed with the district court that Accolade's creation of copies and translations during the reverse engineering process constituted infringement under the literal terms of the Copyright Act.[125]   However, it held that such reverse engineering constituted a fair use when disassembly was the only way to gain access to the ideas and functional requirements for compatibility, which were not protected by copyright.[126]

The court analyzed the alleged fair use defense based on the four factors of § 107.[127] With regard to the first fair use factor, "the purpose and character of the use," the court examined whether the copying executed during decompilation was for a commercial purpose.[128] Although it held that since the copying was for a commercial purpose, it weighed against a finding of fair use, such presumption of unfairness could be rebutted by the characteristics of a particular commercial use.[129] The purpose of the reverse engineering by Accolade was solely to discover the functional requirement for its game cartridges to be compatible with the Sega Genesis. Thus, the court said, the immediate purpose of copying was not commercial.[130]

---

    122.   Sega also asserted claims for trademark infringement and false designation of origin in violation of the Lanham Act, 15 U.S.C. §§ 1114(1)(a), 1125(a) (1988). *Id.* Arguing that the TMSS enabled Sega to falsely pass off Accolade's games as its own, Accolade counterclaimed for false designation of origin. *Id.*
    123.   785 F. Supp. 1392, 1402 (N.D. Cal. 1992).
    124.   *Sega*, 977 F.2d at 1514.
    125.   *See id.* at 1518-20.
    126.   *See id.* at 1523-28.
    127.   *Id.* at 1521-22.
    128.   *See id.* at 1522.
    129.   *Id.*; *see also* Hustler Magazine, Inc. v. Moral Majority, Inc., 796 F.2d 1148, 1152 (9th Cir. 1986); Maxtone-Graham v. Burtchaell, 803 F.2d 1253, 1262 (2d. Cir. 1986) (holding that "[t]he commercial nature of a use is a matter of degree, not an absolute"); Twin Peaks Prod. v. Publ'ns Int'l, Ltd., 996 F.2d 1366, 1374-75 (2d Cir. 1993). Few of the reported cases involved nonprofit educational purpose. Most of the uses involved some degree of direct or indirect monetary gain. *See* PATRY, *supra* note 83, at 731.
    130.   The Ninth Circuit Court relied on § 102(b) of the Copyright Act. *Sega*, 977 F.2d at 1524.

With regard to the second factor, "the nature of the copyrighted work," the court noted that not all copyrighted works receive the same level of protection.[131]   Original works by an author are due more protection than the ideas or functional aspects underlying the work. The court stated that a computer program is a hybrid of both functional components and protectable expression given the primarily utilitarian nature of computer programs.[132]   Since Sega's games contained unprotected aspects that could not be studied without copying them in the decompilation process, they are afforded a lower level of protection.[133]

The court conceded that the third statutory factor, "the amount and substantiality of the taking," weighed against Accolade because it disassembled and thereby made intermediate copies of entire Sega game programs.[134] However, the court remarked that the copying of an entire program does not preclude a fair use defense per se.[135] In fact, where the use of the copied material is as limited as it was in Accolade's instance, it carries very little weight.[136]

With regard to the fourth statutory factor, "the effect of the use on the market for or value of the copyrighted work," the effect on the potential market for the copyrighted work was not Accolade's intent to rob Sega of any of its market share for its games, but to become a legitimate competitor by producing Genesis compatible games.[137] Many Genesis owners have more than one game, so the existence of Accolade's products would not diminish the market potential for Sega. Additionally, any attempt to create a monopoly by making it impossible for other companies to compete is in opposition to the Copyright Act's goal of promoting creative expression.[138]

The court concluded its copyright infringement analysis by stating that "where 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."[139]

---

131. *Id.* at 1523.
132. *Id.* at 1524.
133. *Id.* at 1525-26.
134. *Id.* at 1526.
135. *See id.*
136. *Id.* at 1527.
137. *Id.* at 1523.
138. *Id.* at 1523-24.
139. *Id.* at 1527-28.

3.   *Atari v. Nintendo*

The United States Court of Appeals for the Federal Circuit in *Atari Games Corp. v. Nintendo of America, Inc.* (Atari) took another step forward to weaken a copyright owner's protection against the use of reverse engineering.[140] It explicitly stated that a software developer could lawfully reverse engineer the copyrighted work in order to understand the unprotectable ideas in a computer program, unless the copy from which the reverse engineering was done was an unauthorized copy.[141]   The purpose of the underlying reverse engineering in *Atari* was the same as that in *Sega*:   making compatible games with a copyrighted work. However, part of Atari's reverse engineering process involved fraud on the Copyright Office.[142]   Thus, the Federal Circuit upheld the district court's issuance of a preliminary injunction barring Atari's distribution of its competing video game based both on infringing intermediate copying and on a determination that Atari's end product was substantially similar to the allegedly infringed work.[143]

Nintendo manufactured a home video game system, the NES, which includes a monitor, console, and controls.[144] The console was a base unit into which a user inserted game cartridges which contained game programs for the NES.[145] Nintendo created a security program, the 10NES, which prevented the NES from accepting cartridges which were not manufactured or licensed by Nintendo.[146]   Both the console and authorized game cartridges contained microprocessors programmed with the 10NES, a "master chip" or "lock" microprocessor in the console and a "slave chip" or "key" microprocessor in the game cartridges.[147] When the cartridge was inserted into the console, the two 10NES programs exchanged a series of values based on an initial, randomly selected number.[148] If the final digits of the two series were equal, the console was unlocked, and a user could proceed to play the game.[149] If a user inserted

---

140.   975 F.2d 832 (Fed. Cir. 1992).
141.   *See id.* at 843.
142.   *See id.* at 836.
143.   *See* Atari Games Corp. v. Nintendo of Am., Inc., 18 U.S.P.Q.2d 1935, 1939-40 (N.D. Cal. 1991), *aff'd,* 975 F.2d 832 (Fed. Cir. 1992).
144.   *See Atari Games*, 975 F.2d at 835.
145.   *Id.*
146.   *See id.* at 836.
147.   *Id.*
148.   *Id.*
149.   *See* Atari Games Corp. v. Nintendo of Am., Inc., 30 U.S.P.Q.2d (BNA) 1401, 1403 (N.D. Cal. 1993).

an unauthorized cartridge, the "key" could not open the "lock," and the game did not function.[150]

Atari first attempted to analyze and replicate the NES security system in 1986.[151] After Atari found that it could not break the 10NES program by monitoring the communication between the console and cartridge chips, it tried to break the code by analyzing the chips.[152] Still, Atari could not read the code sufficiently to replicate the NES security system.

In early 1988, counsel for Atari then filed an application with the Copyright Office for a copy of the 10NES program, stating that they needed the code because Atari was a defendant in infringement litigation in the Northern District of California.[153] Since no lawsuit had yet been filed against Atari, that declaration was false.[154] Using this source code, Atari was able to decipher the NES program and develop its own program, the Rabbit program, to unlock the NES.[155] Since Atari chose a different microprocessor and programming language, the line-by-line instructions of the 10NES and Rabbit programs vary.[156] Nonetheless, the Rabbit program generated signals functionally indistinguishable from the NES program, thereby allowing owners of Atari cartridges to play games on Nintendo's system.[157]

Nintendo filed suit for copyright and patent infringement.[158] Atari sought in a separate motion to enjoin Nintendo's alleged antitrust violations and alleged misuse of its property rights.[159] In support of its motion for a preliminary injunction, Nintendo argued that both Atari's end product and its intermediate copying of the 10NES program during

---

150. *See id.* at 1410-11.
151. *See Atari Games*, 975 F.2d at 836.
152. *Id.*
153. *See id.*
154. *Id.*
155. *Id.*
156. *Id.*
157. *See id.* at 837.
158. Atari Games Corp. v. Nintendo of Am., Inc., 897 F.2d 1572, 1574-75 (Fed. Cir. 1990). First, Nintendo sent letters to Atari's retailers threatening suit if sales of the unauthorized games continued, and Atari sued Nintendo for antitrust violations and unfair competition. *See id.* Nintendo's subsequent copyright and patent infringement lawsuit was consolidated with Atari's. *Id.* at 1575. Atari requested, and the district court granted, a preliminary injunction barring Nintendo from suing retailers of Atari's NES-compatible games. *Id.* However, the Court of Appeals for the Federal Circuit reversed, ruling that Atari had not adduced sufficient facts on the issue of Nintendo's allegedly anticompetitive conduct. *Id.* at 1577-78. Atari's antitrust counterclaims and its related copyright and patent misuse defenses were subsequently severed for separate trial following trial of Nintendo's infringement claims. *See* Rex Bossert, *Nintendo Is Victorious in Patent Claim Against Foe*, S.F. DAILY J., July 30, 1993, at 1, 7.
159. *See Atari*, 18 U.S.P.Q.2d (BNA) at 1938.

the reverse engineering process infringed the 10NES copyright.[160]   In response, Atari argued that it had copied, and taken, only functional elements unprotected by copyright.[161]   The district court entered a preliminary injunction against Atari.[162]   It ruled that even if the doctrine of merger excused some similarities between the Rabbit and 10NES programs, Atari had taken more steps than necessary to achieve interoperability.[163]

   The Federal Circuit affirmed the preliminary injunction and held that Atari's use of the illegally obtained copy of the 10NES program barred the application of the fair use doctrine.[164]   However, the court disagreed with the district court's assumption that reverse engineering was not a fair use.[165]   The Federal Circuit stated that copying associated with reverse engineering was not infringement per se.[166]   It held that "reverse engineering object code to discern the unprotectable ideas in a computer program is a fair use."[167]   The underlying policy objectives of the Copyright Act were not to reward authors, but to promote the progress of science and encourage authors to share their works.[168]

4.   Comparative Analysis of *Sega*, *Atari*, and *Sony*

   Since the Ninth Circuit's decision in *Sega*, some criticism has been directed at the fair use analysis adopted by the court.[169]   Nevertheless, many other commentators have portrayed the decision as progressive legal thought.[170]   From the viewpoint of the original computer software creator, if wholesale duplication by a competitor who produces a competing product is allowed, it would deduce directly from the code a number of possible trade secrets, such as mathematical formulas, data structures, program and module organization, as well as compression and

---

   160.   *See id.*
   161.   *Id.*
   162.   *See id.* at 1935.
   163.   *Id.* at 1938-39.
   164.   Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 840 (Fed. Cir. 1992).
   165.   *Id.* at 843.
   166.   *Id.*
   167.   *Id.*
   168.   *See id.* at 842.
   169.   *See* Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977 (1993).
   170.   *See, e.g.*, Dennis S. Karjala, *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller*, 19 U. DAYTON L. REV. 975 (1994); S. Carran Daughtrey, *Reverse Engineering of Software for Interoperability and Analysis*, 47 VAND. L. REV. 145, 172-81 (1994); Pamela Samuelson, *Fair Use for Computer Programs and Other Copyrightable Works in Digital Form: The Implications of* Sony, Galoob, *and* Sega, 1 J. INTELL. PROP. L. 49 (1993).

other algorithms.[171]   Indeed, the decompilation during the process of reverse engineering technically constitutes infringement of the § 106(1) exclusive right of reproduction. However, the intermediate copying that is indispensably required during the decompilation is not unlawful. Computer programs, unlike traditional copyrighted works, do not bear unprotected ideas on their face because vendors distribute software in object code, which is nonhuman-readable.[172]   Opponents of reverse engineering attempt to protect the utilitarian functionality rather than expression. Thus, if the decompilation is supported by the fair use defense under reasonable circumstances, such trade secrets are no longer secrets because they are discoverable by lawful means.

    The opponents also argue that the inappropriate extension of the fair use doctrine into an area in which it does not belong could leave original software developers powerless to stop the use of their software by others to reap profits that would otherwise belong to the original authors.[173] In contrast, from the view of consumers and independent software developers, without application of the fair use doctrine into the area of reverse engineering, original authors may be able to receive patent-like protection for functional elements of the computer programs which are not directly observable.

    This Article argues that decompilation associated with intermediate copying during the reverse engineering process should be privileged by fair use on limited conditions. To prove such an argument, it discusses the nature of a computer program, the particular characteristics of the process of decompilation, and public policy regarding goals of copyright law and dynamic competition. The Ninth Circuit's recent milestone decision in *Sony Computer Entertainment, Inc. v. Connectix Corp.* (*Sony*),[174] has strongly confirmed the argument permitting reverse engineering of software programs for the purpose of examining their unprotected elements. After a comparative analysis of the current key cases regarding fair use, this Article attempts to draw a clear line of distinction between copyright infringement and legally pure reverse engineering when determining fair use.

---

    171.   *See* BERNARD A. GALLER, SOFTWARE AND INTELLECTUAL PROPERTY PROTECTION: COPYRIGHT AND PATENT ISSUES FOR COMPUTER AND LEGAL PROFESSIONALS § 9, at 106 (1995).
    172.   *See* GOLDSTEIN, *supra* note 78, at 85-91.
    173.   *See* Derek Prestin, *Where to Draw the Line Between Reverse Engineering and Infringement:* Sony Computer Entertainment, Inc. v. Connectix Corp., 3 MINN. INTELL. PROP. REV. 137, 139 (2002).
    174.   203 F.3d 596 (9th Cir. 2000).

a.   Facts of the *Sony* Case

Sony Computer Entertainment, Inc. (Sony) brought this copyright infringement action against Connectix Corporation (Connectix). Sony develops and markets the Sony PlayStation, a video game system. The Playstation consists of console and controllers and plays games that are inserted into the PlayStation on compact discs.[175] Sony owns the copyright on the basic input-output system (BIOS), which is the software program that operates its PlayStation.[176]

Connectix, a software manufacturer, makes and sells a software program called "Virtual Game Station" (VGS). To develop its system, Connectix emulated the PlayStation's hardware and operating system as well as its BIOS so that PlayStation games could be played on personal computers.[177] Thus, computer owners who buy the VGS software are able to play Sony PlayStation games on their computers without the support of PlayStation's hardware. Connectix engineers repeatedly copied Sony's BIOS into the random access memory of their personal computers during the process of reverse engineering in order to develop a functional PlayStation emulator.[178] The VGS, however, does not contain any of Sony's copyrighted material.[179] Sony claimed infringement and sought a preliminary injunction.

The district court granted the motion, enjoining Connectix from (1) copying or using Sony BIOS code in the development of the Windows compatible VGS and (2) from selling such stations compatible either with Windows or Macintosh.[180] The court reasoned that Connectix's intermediate copying of Sony BIOS code, in order to develop emulation software, was not a fair use under 17 U.S.C. § 107.[181] Connectix appealed, and the Ninth Circuit reversed the district court's order of injunction and held that "[t]he intermediate copies made and used by Connectix during the course of its reverse engineering of the Sony BIOS were protected fair use, necessary to permit Connectix to make its non-infringing Virtual Game Station function with PlayStation games."[182]

---

175.   *See id.* at 598.
176.   *Id.*
177.   *Id.*
178.   *See id.* at 596.
179.   *See id.* at 598.
180.   *See* Sony Computer Entm't, Inc. v. Connectix Corp., 48 F. Supp.2d 1212, 1224 (N.D. Cal. 1999).
181.   *Id.* at 1212.
182.   *Sony*, 203 F.3d at 599.

b.   Uniqueness of Computer Programs

Computer programs have characteristics different from those of other literary works.   First, computer programs are inherently characterized as functional works that directly cause machine processes to be performed.[183] They are programmed for the purpose of being used on a computer or other hardware, unlike the purpose of most other literary works which are meant to be read. Thus, if the functionality of a program is protected too strongly, there is a risk that a copyright owner could obtain a patent-like monopoly, not only over the copyright-protected portions of the program but also over unprotected ideas and methods.[184]   In order to protect processes or methods of operation, the software developer must look to patent laws rather than copyright laws and show patent requirements.[185]

Computer programs are also unique in terms of the form in which they are distributed for public use; the object code form in which computer programs are publicly distributed is unreadable by human beings.[186]   Unlike other forms of literary expression, such as books, computer programs cannot be simply opened and read or examined.[187] Thus, unless the program is investigated and translated into an intelligible form by human beings, which may require copying the copyrighted material, end-users cannot study, research, or even perceive the ideas, processes, structures, or actual methods of operation of the program.[188] Professor Dennis Karjala stated that

> [i]f decompiling publicly distributed but human-incomprehensible object code is considered infringement, the public never receives its fair part of the copyright bargain: a seventy-five-year limited monopoly in exchange for the opportunity to read and be inspired by the work and a free license to copy all of the ideas and processes embodied in the program.[189]

c.   Inevitable Copy of Entire Object Code

As explained previously, most methods of reverse engineering are normally associated with intermediate copying of part or all of the

---

   183. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1525 (9th Cir. 1992). Other copyrightable works like architectural works are protected under the Copyright Act where the design elements are not functionally required. *See* H.R. Rᴇᴘ. Nᴏ. 101-735, 101st Cong., 2d Sess. (1990).
   184. *See* Karjala, *supra* note 170, at 993.
   185. Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 842 (Fed. Cir. 1992).
   186. *See id.* at 832.
   187. *See* Lande & Sobin, *supra* note 18, at 240.
   188. Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 602 (9th Cir. 2000).
   189. Karjala, *supra* note 170, at 994.

original software. Especially when it comes to reverse engineering of the functional elements of programs, which are not visible to users on the computer screen, decompilation as a part of the reverse engineering process indispensably requires making copies of the programs. Thus, translation of the program from object code into source code cannot be accomplished without making copies of the original code.[190]

     If a software developer wants to discern or detect such unprotected ideas, processes, or interface information for the purpose of interoperability with original software, its entire object code must be disassembled in most instances into human readable form. Software reverse engineering is a difficult and time consuming process.[191] The following are the most difficult problems in reverse engineering a computer program: the reverse engineer has no clue as to the higher levels of abstraction; why the code is written in a particular manner; and the processing sequence.[192] The reverse engineer cannot know whether all steps necessary for interoperability have been located without checking the entire program, since "[t]he object-coded representation of a computer program produced by a decompiler lists program steps in the order in which they are coded, not the order in which they are executed."[193] For example, suppose there is a "jump subroutine" instruction at location 0001, which tells the computer not to execute the following instruction at 0002, but to "jump" out of sequence to a completely different part of the program. In this case, the program would jump to a section that the engineer has not yet disassembled, and, therefore, the purpose would be unknown.[194] For this reason, the reverse engineer must decompile the entire program to understand the instructions necessary to achieve interoperability.

     As was the case for Connectix in *Sony*, without checking Sega's entire program, Accolade could not know whether it had all the information necessary to produce Genesis-compatible games. While Accolade copied the entire Sega program, it eventually used only a minor

---

     190.   Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1526 (9th Cir. 1992).

     191.   A modern program may consist of at least 300,000 instructions. Assuming the engineers would take only 30 seconds to decode an instruction, this means that reverse engineers would take 2,500 hours (ten months) to complete the disassembly, and that would only tell them what the raw instructions were. Indeed, one of the most significant obstacles is that they would still have no high level understanding of the code itself. *See* Johnson-Laird, *supra* note 36, at 873.

     192.   Johnson-Laird, *supra* note 36, at 875.

     193.   Cohen, *supra* note 33, at 1125.

     194.   Johnson-Laird, *supra* note 36, at 875-76, 878 ("Any literary work that has twenty to thirty levels of indirection, referring the reader from paragraph 102 to paragraph 239, then to paragraph 97 and then to paragraph 3 and so on, would be deemed to be totally incomprehensible. Yet, this is completely normal in software.").

portion in its product.[195]   Accordingly, decompiling the entire Sega Genesis program was not an immoderation, but a fair use.

    This characteristic of reverse engineering for software significantly provides a legitimate ground for the third statutory factor of the fair use test: the amount and substantiality of the portion used in relation to the copyrighted work as a whole.[196] In *Sony*, Connectix disassembled parts of the Sony BIOS and copied the entire Sony BIOS including protected and unprotected elements multiple times.[197] Such a factor seems to favor Sony; however, as explained above, making copies of a protected program is inevitably necessary to understand how the software functions. It is important to note that no more was copied than was necessary. More importantly, Connectix's end product did not encompass any code of Sony BIOS even though the entire program was copied. Therefore, this factor is of very little weight to preclude a finding of fair use.[198]

d.   The Purpose and Character of the Use of Copyrighted Software

    The first statutory factor of the fair use defense should not be confined only to one relevant fact such as commercialism, but should rather lie in the key inquiry that the use advances the goal of copyright law. The following three factors are vital to an analysis of this inquiry: (1) whether the use is transformative, (2) whether the use is commercial, and (3) whether the alleged infringer's conduct is proper.[199] The approach of the Ninth Circuit in *Sega* did not provide a clear distinction regarding whether the use of information obtained from the original software by using reverse engineering technology was commercial in nature or not-for-profit. Meanwhile, although the court in *Sony* adopted the first two sub-factors, it did not sufficiently crystallize its analysis.

    The key investigation of the first test is to find whether and to what extent the new work is "transformative." The Supreme Court, in *Campbell v. Acuff-Rose Music, Inc.*, developed the productive use theory,[200] stating that an infringement is transformative if the new product

---

    195.   *Sega*, 977 F.2d at 1514-15.
    196.   17 U.S.C. § 107 (2000).
    197.   Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 606 (9th Cir. 1992).
    198.   *Sega*, 977 F.2d at 1526-27.
    199.   *See* Penelope v. Brown, 792 F. Supp. 132, 136 (D. Mass. 1992); Rubin v. Brooks/Cole Publ'g Co., 836 F. Supp. 909, 917 (D. Mass. 1993).
    200.   *See* Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 562 (1985); Folsom v. Marsh, 9 F. Cas. 342, 348 (C.C.D. Mass. 1841) (No. 4,901); Leval, *supra* note 88, at 1111.

did not merely supplant the original work but instead added something new, with a further purpose or different character, thereby altering the first with new expression, meaning, or message.[201] Application of the transformative use standard is important when it stimulates creativity for public benefits and enriches society.

    In *Sony*, the Ninth Circuit recognized the importance of the *Campbell* approach in analyzing reverse engineering contexts. Yet, the court missed a point in its analysis regarding the distinction between intermediate copying made during the reverse engineering process and the transformative use in the *Campbell* case. The court simply found that Connectix's VGS was reasonably transformative because it was a wholly new product.[202] It created a new platform, which consumers could play with games without a Sony PlayStation console, provided a personal computer with a CD-ROM drive was available.[203] Indeed, despite the similarities in function and screen output between the two products, the VGS did not include any code contained in Sony's BIOS. However, this was not why the plaintiff brought the case. The Supreme Court used the words "transformative use," which meant that the thing accused of being copied was transformative. In contrast, the thing accused of being copied in *Sony*, the "intermediate copy," was not transformative.

    The parodic use of the old song in *Campbell* was transformative because the defendant never took the old song and made copies that were exactly the same as the original; rather, it criticized the song and added something new to it. Yet, the intermediate copies made by Connectix during the course of the reverse engineering of the Sony BIOS were not transformative at all; reverse engineering involves only copying.

    Nevertheless, this does not necessarily mean that the reverse engineering is not fair use. The need for reverse engineering to ensure interoperability is so strong that the role of intermediate copying may justify the infringement. Reverse engineering generally serves to promote copyright's ultimate purpose by encouraging creative activities and preventing new works from being unjustifiably denied entry into society.[204] Hence, the role of reverse engineering has certain trans-

---

    201.   Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 579 (1994).
    202.   *Sony*, 203 F.3d at 606.
    203.   *Id.* On this point, the purpose of decompilation by Connectix—which was to make a competing platform (horizontal compatibility)—was different from that of Accolade's decompilation aimed at making games which are compatible with Sega's game platform (vertical compatibility). *See* Gary R. Ignatin, *Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility*, 140 U. PA. L. REV. 2042-44 (1992).
    204.   John A. Williams, *Can Reverse Engineering of Software Ever be Fair Use? Application of Campbell's "Transformative Use" Concept*, 71 WASH. L. REV. 255, 280 (1996).

formative characteristics. The justification turns primarily on the similarity (or transformative nature) of the defendant's final product, if the *Campbell* concepts would be considered expansive under the unique circumstances of software reverse engineering.

In *Sega*, Accolade also created its own transformative product. It had made copies of Sega's video game programs during the process of reverse engineering solely to determine the requirements for compatibility with Sega's game platform.[205] Accolade then developed its own games to operate on the Sega platform. Thus, Accolade's productive use of Sega's copyrighted material was favorable for a finding of fair use.

It was clear that Connectix intended to use Sony's copyrighted material for a commercial purpose, which was to develop a product compatible with games designed for the Sony PlayStation.[206] Thus, this factor favors Sony. However, the commercial exploitation of the competing product should not automatically convert the practice of reverse engineering into a prohibited act.[207] The court rejected the presumption that commercial purpose in copying the copyrighted material gave rise to a presumption of unfairness.[208] In *Sony*, the court held that Connectix's commercial purpose of use was only a "separate factor that tend[ed] to weigh against a finding of fair use."[209] The Supreme Court announced in *Harper & Row Publishers, Inc. v. Nation Enterprises* that "[t]he crux of the profit/nonprofit distinction is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material without paying the customary price."[210] Since Connectix's commercial use of the copyrighted material was an intermediate one, such use was only indirect or derivative.[211] Therefore, Connectix's commercial use should weigh less heavily against a finding of fair use.

---

    205.   Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1514-15 (9th Cir. 1992).
    206.   Connectix's engineers admitted that they disassembled Sony's code not just to study the concepts, but to actually use that code in the development of the Virtual Game Station. *See* Sony Computer Entm't, Inc. v. Connectix Corp., 48 F. Supp. 2d 1212, 1220 (N.D. Cal. 1999).
    207.   *See* Roy Exp. Co. Establishment v. Columbia Broad. Sys., Inc., 503 F. Supp. 1137, 1144 (D.C.N.Y. 1980); Meeropol v. Nizer, 560 F.2d 1061, 1069 (2d Cir. 1977) ("[T]he mere fact that . . . for commercial gain, does not, standing alone, deprive . . . of the fair use defense."); GOLDSTEIN, *supra* note 78, at 89, 91.
    208.   Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 584, 594 (1994).
    209.   Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 606 (9th Cir. 1992) (citation omitted).
    210.   Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 562 (1985).
    211.   *Sony*, 203 F.3d at 607; Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1522 (9th Cir. 1992).

In *Sega*, the fact that Accolade reverse engineered Sega's software to produce a product that was compatible with the Genesis console favored Sega. However, Accolade's commercial use was for a legitimate, essentially nonexploitative purpose, and, thus, such use could be described as having minimal significance.[212]

The third factor, whether the conduct was proper, asks whether the original was copied in good faith to benefit the public.[213] Like Accolade, Connectix copied the protected work in good faith; they copied the work solely to understand the functional elements for compatibility with the original products and to develop their own products. Since the functional information was not available to the public and direct requests to the copyrighted owners for the information failed, Accolade and Connectix had no other way to access this information except through reverse engineering of the copyrighted material. Although the second factor disfavors Connectix and Accolade, upon weighing these factors, the purpose and character of the defendants' use favored a finding of fair use.

e.   Effect of the Use upon the Potential Market

Under the fourth fair use factor, the effect of the use upon the potential market for or value of the copyrighted work, the courts should test not only the extent of market harm caused by the particular actions of the alleged infringer, but also "'whether unrestricted and widespread conduct of the sort engaged in by the defendant . . . would result in a substantially adverse impact on the potential market' for the original."[214] The likelihood of market harm is presumed when there is direct duplication for a commercial purpose.[215] A work that merely supplants or supersedes another is likely to have a substantially adverse impact on the potential market for the original.[216]

---

212.   *Sega*, 977 F.2d at 1523.

213.   *Harper & Row Publishers*, 471 U.S. at 562; Rubin v. Brooks/Cole Publ'g Co., 836 F. Supp. 909, 918-19 (D. Mass. 1993). The first statutory factor of fair use turns on whether the original was copied in good faith to benefit the public or primarily for the commercial interests of the infringer. A defendant's conduct in tearing the copyright mark off of a plaintiff's notecard before sending it to the Italian artisans was bad faith. 4 NIMMER & NIMMER, *supra* note 67, § 13.05[A][1][d].

214.   Campbell v. Acuff-Rose, 510 U.S. 569, 590 (1994) (quoting 4 NIMMER & NIMMER, *supra* note 67, § 13.05[A][4]); *Harper & Row*, 471 U.S. at 568. The inquiry "must take account not only of harm to the original but also of harm to the market for derivative works." *Harper & Row*, 471 U.S. at 568.

215.   *Acuff-Rose*, 510 U.S. at 591.

216.   *See Sony Corp.*, 464 U.S. at 451; Folsom v. Marsh, 9 F. Cas. 342, 348 (C.C.D. Mass. 1841) (No. 4,901).

No such presumption or inference of market harm, however, is applicable to a case involving something beyond mere duplication for commercial purposes.   For example, when the end product is transformative, it is less likely to cause market harm.[217]   Even if there might be market harm, this would be legitimate harm in a competitive marketplace. As we can see, the fourth fair use factor bears a close relationship to the purpose and character inquiry of the first factor.

In *Sony*, the court held that since the Virtual Game Station is transformative, and does not merely supplant Sony's PlayStation console, Connectix's product is a legitimate competitor in the market for platforms on which Sony and Sony-licensed games can be played.[218]   In *Sega*, Accolade did not attempt to replace Sega's release of any particular games, but sought only to become a legitimate competitor by creating a transformative product.[219]   It runs counter to the statutory purpose of promoting creative expression if a company is allowed to monopolize the market by making it impossible for others to compete.[220]

f.   Public Policy Analysis

As will be discussed in greater detail in Part II.D, prohibiting any decompilation for the purpose of achieving interoperability or compatibility with the protected work has a conceivable problem in light of competition policy.   If software reverse engineering automatically constitutes an unfair use, the copyright owner, whose product becomes the industry standard, creates a de facto monopoly on the functional aspects of a work.[221]   Indeed, not only can it easily eliminate its competitors, whose fundamental goal is imperative to interoperability because they are locked into the product, but it can also build a barrier against new entities to relevant and potential markets. The court in *Sega* reasoned that because such protection is expressly denied under § 102(b) of the Copyright Act, it is only available if the more stringent requirements of the patent laws are met.[222]

The policy allowing independent developers to reverse engineer computer software under limited circumstances results in public benefits. The court held that Accolade's identification of the functional

---

217. *See* Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 607 (9th Cir. 1992).

218. *Id.*

219. *See* Sega Enters., Ltd. v. Accolade, Inc., 977 F.2d 1510, 1523 (9th Cir. 1992).

220. *Id.* at 1523-24.

221. *Id.* at 1526.

222. *Id.*; *see also* Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 159-64 (1989).

requirement for Genesis compatibility has led to an increase in the number of independently designed video game programs offered for use with the Genesis console.[223] It also noted growth of creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, which the Copyright Act was intended to promote.[224] The ultimate aim of the Copyright Act is to benefit the public through the distribution of knowledge, rather than to reward individual authors.[225]

    After the first sale of a protected product having strong marketing potential, the user becomes locked into the product unless compatible products are produced.[226] The standard owner can easily obtain significant competitive advantages, and eliminate its competitors by slightly changing its product with little or no notice, even though competitors independently achieve compatibility with the standard without reverse engineering. Such slight modifications "suddenly make competitors' products no longer compatible; the industry [subsequent entrepreneur] again finds itself struggling to maintain any market share that it had gained through the development of compatible components and software."[227] Accordingly, the competition factors used to resolve the reverse engineering rights do not reduce the opportunity for using copyright as an exclusionary strategy. Even though copyright law grants an author certain exclusive rights, the author should not use these rights to maintain its dominance and control in a manner adverse to the public policy.[228]

    With regard to this competition policy discussion, this Article also emphasizes the importance of "network effects." The network effects imply that the more consumers own identical or compatible goods, the

---

    223.   *Sega*, 977 F.2d at 1523.
    224.   *Id.*
    225.   *See* Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 432, 447 (1984).
    226.   The European Commission brought suit against IBM for an abuse of a dominant market position.   *See* Int'l Bus. Machs. v. Commission, 1981 E.C.R. 1857; Joseph Farrell, *Standardization and Intellectual Property*, 30 JURIMETRICS J. 35, 38 (1989); Linda G. Morrison, *The EC Directive on the Legal Protection of Computer Programs: Does It Leave Room for Reverse Engineering Beyond the Need for Interoperability?*, 25 VAND. J. TRANSNAT'L L. 293, 304 (1992).
    227.   *See* Morrison, *supra* note 226, at 305.
    228.   *See* Stephen Shankland, *Lawyer Lessig Raps New Copyright Laws*, CNET News.com. (Aug. 29, 2001) (stating that "[c]opyright and patent law, ostensibly designed to protect innovation, now have become tools large companies can use to maintain their dominance and control").

more the value of a good increases.[229]   As a result, each consumer benefits from service, accessibility and other suppliers' networks.   As cases illustrate, a central problem is that effective competition requires that the games or game players tend to be programmed toward a dominant entity (e.g., Sega Genesis game system, Sony PlayStation, Nintendo, etc.).   Hence, to be effective in competing either in the programming of games, platform programs, or hardware to run such things, the competitor needs to have a product that is compatible with an existing system.   On this point, a number of legal commentators and economists have insisted on the legality of reverse engineering in network industries.[230]

This Article suggests that statutory clarification of a reverse engineering right is necessary to keep up with the fundamental goal of copyright law.   This is in spite of some efforts on the part of Congress to resolve emerging conflicts between copyright protection and public rights to access uncopyrighted components.   The legislation alleviated some of the conflicts in the field of reverse engineering for computer chips.   Congress passed the Semiconductor Chip Protection Act of 1984 (SCPA), which permits reverse engineering, in some limited circumstances, to reproduce an image or template, or mask work that is used to create the circuitry on a silicon chip.[231]   Thus, dissemination of

---

229. *See* Joseph Farrell & Garth Saloner, *Standardization, Compatibility, and Innovation,* 16 RAND J. ECON. 70, 70 (1985) (stating that a consumer's value for a good increases when there is a compatible good).

230. *See* Lemley & McGowan, *supra* note 17, at 525; Jeffrey Church & Roger Ware, *Network Industries, Intellectual Property Rights and Competition Policy, in* COMPETITION POLICY, INTELLECTUAL PROPERTY RIGHTS IN THE KNOWLEDGE-BASED ECONOMY (1998); William E. Cohen, *Competition and Foreclosure in the Context of Installed Base and Compatibility Effects,* 64 ANTITRUST L.J. 535, 550 (1996).

231. The Act provides:

(a)   Notwithstanding the provisions of section 905, it is not an infringement of the exclusive rights of the owner of a mask work for

   (1)   a person to reproduce the mask work solely for the purpose of teaching, analyzing, or evaluating the concepts or techniques embodied in the mask work or the circuitry, logic flow, or organization of components used in the mask work; or

   (2)   a person who performs the analysis or evaluation described in paragraph (1) to incorporate the results of such conduct in an original mask work which is made to be distributed.

(b)   Notwithstanding the provisions of section 905(2), the owner of a particular semiconductor chip product made by the owner of the mask work, or by any person authorized by the owner of the mask work, may import, distribute, or otherwise dispose of or use, but not reproduce, that particular semiconductor chip product without the authority of the owner of the mask work.

17 U.S.C. § 906 (2000).

the ideas embodied in a mask work may be exempt from infringement liability despite proof of unauthorized copying and striking similarity. If the purpose of reverse engineering is to analyze, study and understand an existing chip, and the resulting chip product contains technological improvement, the reverse engineering is not an infringement.[232]

Another example of the proactive stances protecting reverse engineering right is the exception provision of DMCA, which considers decompilation of programs lawful. Yet, the exception limits the purpose of reverse engineering to only interoperability. Indeed, the DMCA grants copyright owners extra technical protection for emerging digital technologies. Thus, a reverse engineer cannot bypass technical protections other than when necessary to achieve program-to-program interoperability. This places many legitimate decryption activities at risk: reverse engineers may be subject to violation of DMCA even though they can eschew liability under the Copyright Act.

Uncertainty is more serious when it comes to a contractual license (e.g., a shrink-wrap license) that can impose much more restrictive terms on reverse engineering. Therefore, the anticircumvention provisions and UCITA could have the effect of prohibiting a buyer or licensee from exercising the reverse engineering rights deemed inconsistent with the provisions or private contracts, including the rights on the development of a compatible product, error correction, creative activities, etc.

g.   Clarification of Reverse Engineering Rights

In general, reverse engineering of a computer program for various purposes should be permitted if the end product does not include protected features of the original program. A person who has a right to use a copy of a computer program is allowed, without a copyright holder's consent, to study, research, or test the functioning of the program in order to determine the ideas and principles or to criticize errors in the program.[233] For example, decompilation for error correction could be justified in cases where the copyright owner is not able to fix the errors within a reasonable time or at a reasonable price, or when the owner has gone out of business.[234]

---

    232.   Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 842 n.5 (Fed. Cir. 1992); GALLER, *supra* note 171, at 107-08.

    233.   Council Directive 91/250/EEC, art. 5(3), 1991 O.J. (L. 122) 42; 17 U.S.C. § 107 (2000).

    234.   Council Directive 91/250/EEC, art. 5(1), at 42 ("In the absence of specific contractual provisions, the acts . . . shall not require authorization by the rightholder where they are necessary for the use of the computer program by the lawful acquirer in accordance with its intended purpose, including for error correction."); Brian Fitzgerald, Cristina Cifuentes, Anne Fitzgerald &

Since the ideas or interface information on computer software are not easily detected, decompilation of the object code must be allowed to obtain the information. Decompilation, the purpose of which is to achieve interoperability or compatibility, is indispensably involved in copying the original software. Reverse engineering for interoperability or compatibility should be given the green light for the purpose of copyright law if interoperability or compatibility meets the following conditions. First, a software developer should make the effort to obtain information about the interfaces or relevant functional elements from a copyright owner. Second, if this is not successful, then reverse engineering may be permitted. Decompilation of the object code should only be necessary to obtain the required information when there is no other alternative means of accessing such information. This would be a difficult burden for a defendant to establish.

Third, the purpose of reverse engineering is not to engage in exploitation. Rather, it is a creative activity through understanding unprotected functional requirements for interoperability or compatibility. The new product should not merely supersede the original work, but should contain a transformative and extended character. In other words, the final product should be an original work in that it may not infringe on the original computer program.[235] Fourth, the information sought by reverse engineering must be necessary to achieve interoperability or compatibility. Hence, the use of the copyrighted material made during reverse engineering must not exceed what is required to understand the unprotected elements of the program.[236]

Fifth, the final product should contain no code of the original program, or as little as possible of these portions, which are necessary to achieve interoperability or compatibility. We may presume that there is no likelihood of cognizable market harm for the reverse engineered software if final goods do not include any code of the original program. However, since copyright owners' exclusive rights include the right to make derivative works, it is important to determine the scope of derivative work protection.[237] When the defendant's final product is substantially similar to the plaintiff's, this results in a prima facie case of copyright infringement.[238] If, however, the defendant's end product is not

---

Michael Lehmann, *Innovation, Software, and Reverse Engineering*, 18 SANTA CLARA COMPUTER & HIGH TECH. L.J. 121, 146-47 (2001).

235. *See* S. REP. No. 105-190, at 32 (1998).

236. *See Atari*, 975 F.2d at 843.

237. *See* 17 U.S.C. § 106(2) (2000) (stating that a copyright owner has the exclusive right "to prepare derivative works based upon the copyrighted work").

238. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1518-19 (9th Cir. 1992).

substantially similar, attention shifts to the fair use inquiry as a defense for such copying.[239]   As viewed in *Atari*, if the final product is created through the process of software reverse engineering which involves an improperly acquired copyrighted work or fraud perpetrated in the Copyright Office, the reverse engineering gives no sound reason for fair use.[240]

C.   *Reverse Engineering Under the DMCA*

The first congressional awareness of the legitimacy of software reverse engineering was codified in the Digital Millennium Copyright Act of 1998, but the authorization is not sufficient to permit the kind of reverse engineering that the fair use doctrine allows.[241]   Because a substantial portion of copyright industry revenues has come from the mass-market sale of digital forms of copyrighted works, industry groups persuaded Congress to provide legal reinforcements to the technical protection measures protecting those works.[242]   As a result, § 1201, the anticircumvention provision, prohibits someone from circumventing a technological measure that effectively controls access to a copyrighted work.[243]   Thus, if a creator of original software developed a specialized "lock-and-key" tool in its program in order to prevent the reverse engineering of the program, the circumvention of the security measure

---

239.   *See* PATRY, *supra* note 83, at 756.

240.   *See* DSC Communications v. DGI Techs., Inc., 898 F. Supp. 1183 (N.D. Tex. 1995), *rev'd in part*, 166 F.3d 772 (5th Cir. 1999).   The court held that reverse engineering of firmware acquired on the open market constituted a fair use, but that DGI's reverse engineering of improperly acquired operating system software did not constitute a fair use.   *Id.* at 1194.

241.   *See* Digital Millennium Copyright Act, Pub. L. No. 105-304, 112 Stat. 2863 (1998). The Act implemented the World Intellectual Property Organization (WIPO) Copyright Treaty, which requires contracting parties to

> provide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by authors in connection with the exercise of their rights under this Treaty or the Berne Convention and that restrict acts, in respect of their works, which are not authorized by the authors concerned or permitted by law.

WIPO Treaty, Apr. 12, 1997, art. 11, S. Treaty Doc. No. 105-17 (1997); H.R. REP. No. 105-845, at 159 (1998).

242.   *See* Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575, 1630 (2002).

243.   17 U.S.C. § 1201(a)(1)(A) (2000) ("No person shall circumvent a technological measure that effectively controls access to a work protected under this title.").   Section 1201 consists of three separate species of anticircumvention: a basic anticircumvention provision (§ 1201(a)(1)(A)), an antitrafficking provision (§ 1201(a)(2)), and additional violations (§ 1201(b)).   *Id.*

would violate § 1201.[244]   Indeed, if one engages in decompilation of the object code of the copyrighted computer program, which contains a technological measure, one could not exercise a fair use defense to anticircumvention or antitrafficking in § 1201.

The DMCA has a dual control regime that provides protection beyond traditional copyright law.   Section 1201 protects not only a measure controlling access to a work (access control), but also a measure controlling access to a copy of a work (copy control).[245] Access control is a right additionally codified in the DMCA; copy control corresponds to access in the copyright sense of the right to distribute copies of the work.[246] There are differences between access control and copy control. The DMCA bans the circumventing of measures controlling access to a work[247] and the disseminating of devices designed to circumvent access control.[248]   Therefore, no one is allowed to access a work unless a copyright owner provides an access right. However, the DMCA does not prohibit breaking into copy control; it only prohibits trafficking in tools that circumvent copy control.[249]

The statutory language does not make all circumvention of the technological measures illegal. Congress created the reverse engineering exception that was designed to support reverse engineering necessary for achieving interoperability. Section 1201(f)(1)(A) provides:

> Notwithstanding the provisions of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs, and that have not previously been readily available to the person engaging in the circumvention, to the extent any such acts of identification and analysis do not constitute infringement under this title.[250]

---

244.   *See* Jonathan Band & Taro Isshiki, *Peace at Last? Executive and Legislative Branch Endorsement of Recent Software Copyright Case Law*, 16 NO. 2 THE COMPUTER LAWYER. 1, 4 (1999).

245.   17 U.S.C. § 1201.

246.   Copyright owners have always had "access controls" in the literal sense. *See* Harper & Row Publishers, Inc. v. Nat'l Enters., 723 F.2d 195, 201 (2d Cir. 1983); Jane C. Ginsburg, *Copyright Legislation for the "Digital Millennium"*, 23 COLUM.-VLA J.L. & ARTS 137, 140 (1999).

247.   17 U.S.C. § 1201(a)(1).

248.   *Id.* § 1201(a)(2).

249.   *Id.* § 1201(b).

250.   *Id.* § 1201(f)(1)(A).

Specific conditions are set forth in this provision to grant the exception against the basic anticircumvention provision. First, the copy of the computer software that is the subject of the analysis must be lawfully acquired. It is clear that the statutory language "a person who has lawfully obtained the right to use a copy" must include someone who has obtained a copy subject to a license.[251] Otherwise, the statute would have been limited to one "who lawfully owns a copy." A person is able to access a work in order to ensure fair use only when he or she legitimately obtains the work.[252] The computer program must be acquired from a legitimate source, along with any necessary serial codes, passwords, or other means.[253] Purchase of a copy of off-the-shelf software from an ordinary retail store or downloading off the Internet would be sufficient even without getting a special license.

Second, the sole purpose must be to achieve interoperability of an independently created program with other programs. The objective of reverse engineering of a program, including the protective system, should be to identify and extract such elements necessary to achieve interoperability, and as a result, the final product should be a new and original work.[254] The legislative history, however, does not clearly define the scope of interoperability. It simply cites *Sega* as an example of interoperability, which is involved in compatibility with a copyrighted work.[255] A competitor may need to break the code for many purposes other than for seeking to achieve interoperability. For example, reverse engineering can be conducted for error correction (e.g., a repair service provider for a machine corrects error embodied in the diagnostic program)[256]; for developing an independently created program that understands the file and command structure of the original program and can therefore operate on files or documents created using the original program (e.g., a new spreadsheet program that can read files created using Lotus 1-2-3)[257]; and for developing highly specialized software which may not be functionally interchangeable with the original program.

---

251. *See id.*
252. H.R. REP. NO. 105-551, pt. 1, at 18.
253. *See* S. REP. NO. 105-190, at 32 (1998).
254. *See id.*
255. *Id.*
256. *See* MAI Sys. Corp. v. Peak Computer, Inc., 991 F.2d 511, 517 (9th Cir. 1993).
257. MILGRIM, *supra* note 24, § 1.05[5].

If software reverse engineering met such conditions, it could be protected by claiming a fair use.[258] Section 1201(f)(1) is the only reverse engineering exception to § 1201(a)(1)(A), and no reasons for reverse engineering other than that specified in § 1201(f)(1) would suffice as a defense of § 1201(a)(1)(A). Fair use is a defense only in regard to copyright infringement; it is not a defense to § 1201.[259] Because it only deals with access to copyrighted materials that have been stored in a digital format, the DMCA involves what is, strictly speaking, something other than copyright infringement. Thus, if a software developer engages in reverse engineering for a purpose other than interoperability, the developer can still claim fair use as a defense to copyright infringement. However, one could not claim fair use as a defense to anticircumvention. The limitation of the statutory language on reverse engineering in regard to interoperability alone undermines a balance between protecting copyrighted work and accessing unprotected materials. Since a comprehensive ban on circumvention would negate the fair use doctrine,[260] it could create potential liability against software reverse engineers except for interoperability purposes.

Even though § 1201(f) does not intend to change public policy and the effect of prior case law, it very likely has anticompetitive effects in the market. The report of the Judiciary Committee states that "[t]he purpose of this section is to foster competition and innovation in the computer and software industry."[261] The objective of this provision is "to ensure that the effect of current case law interpreting the Copyright Act is not changed by enactment of this legislation for certain acts of identification and analysis done in respect of computer programs."[262] Nevertheless, circumvention that is part of a reverse engineering process for a purpose other than interoperability would be banned even in cases where the legitimate use of the copyrighted work would not violate any of the owner's rights under existing copyright law. Indeed, decrypting a

---

258. *See* David Nimmer, *A Riff on Fair Use in the Digital Millennium Copyright Act*, 148 U. PA. L. REV. 673, 702 (2000) (stating that section 1201(f)(1) "is designed to ensure that the judicial extension of fair use to reverse engineering not be undercut").

259. *See* Universal City Studios, Inc. v. Corley, 273 F.3d 429, 443 (2d Cir. 2001). There has been scholarly criticism of this decision. *See* Cassandra Imfeld, *Playing Fair with Fair Use? The Digital Millennium Copyright Act's Impact on Encryption Researchers and Academicians*, 8 COMM. L. & POL'Y 111, 135-36 (2003); Mihet, *supra* note 93, at 111; Edward Lee, *Rules and Standards for Cyberspace*, 77 NOTRE DAME L. REV. 1275 (2002).

260. *See generally* Digital Future Coalition, *Collected Papers and Press Releases*, http://www.dfc.org/; Julie E. Cohen, *WIPO Copyright Treaty Implementation in the United States: Will Fair Use*, 21 EUR. INTELL. PROP. REV. 236 (1999).

261. S. REP. NO. 105-190, at 32 (1998).

262. *Id.*

protective system generally requires a tool which serves to achieve the decryption effectively as a part of reverse engineering a work. Yet, the tool for reverse engineering is mostly banned because the DMCA strictly restricts publication of information discovered by a reverse engineer.[263] A copyright owner can also provide legal protection against the circumvention of technological measures by imposing contractual limitations.   Thus, the DMCA threatens to deter part legitimate competition by narrowing the scope of the reverse engineering right. This Article suggests that the statutory language be interpreted so as to achieve more open competition and avoid being unduly protective. In terms of competition, reverse engineering of the original program is necessary in order to compete when the central objective is not to overcome the protective system but to allow interoperability. This prospect is considered to be expansive in order to narrow the scope of section 1201's protection. Access to the protective measures for the purpose of exercising fair use rights must be treated as a permissible action in point.

D.   Copyright Misuse and Antitrust Claims

Restrictions   on   software   reverse   engineering   through anticompetitive conduct or copyright license agreements may constitute copyright misuse or a violation of the antitrust laws. The patent misuse defense has long been recognized as an equitable defense against patent infringement based on the patent holder's use of the patent.[264]   In situations where a patentee has improperly extended beyond the physical or temporal scope of the patent grant with anticompetitive effects, such abusive conduct may be prevented as a patent misuse.[265] The judicial concerns regarding the patent misuse doctrine have recently been moving into copyright misuse because the fundamental policies of the patent and copyright laws are similar.[266]

---

    263.   17 U.S.C. § 1201(a)(2), (b)(1) (Supp. V 1999); *see* Samuelson & Scotchmer, *supra* note 242, at 1631, 1658.
    264.   While conduct constituting patent misuse may sometimes be the basis for an affirmative counterclaim, patent misuse in itself is not an actionable tort. *See* B. Braun Med. Inc. v. Abbott Lab., 124 F.3d 1419, 1427 (Fed. Cir. 1997); Senza-Gel Corp v. Seiffhart, 803 F.2d 661, 668 (Fed. Cir. 1986); B. Braun Med. Inc. v. Abbott Lab., 124 F.3d 1419, 1427 (Fed. Cir. 1997); Seungwoo Son, *Selective Refusals to Sell Patented Goods: The Relationship Between Patent Rights and Antitrust Law*, 2002 J. L. TECH. & POL'Y 109, 136-37 (2002).
    265.   Windsurfing Int'l, Inc. v. AMF, Inc., 782 F.2d 995, 1001 (Fed. Cir. 1986) (quoting Blonder-Tongue Labs., Inc. v. Univ. of Ill. Found., 402 U.S. 313, 343 (1971)).
    266.   *See* Lasercomb Am., Inc. v. Reynolds, 911 F.2d 970, 972 (4th Cir. 1990) ("A successful defense of misuse of copyright bars a culpable plaintiff from prevailing on an action for infringement of the misused copyright."); Atari Games Corp. v. Nintendo of Am., Inc., 975

The copyright misuse doctrine was first established in the *Lasercomb America, Inc. v. Reynolds* decision, which expressly upheld the defense against an infringement action.[267] Lasercomb, the copyright owner, included anticompetitive contract provisions in a license agreement which precluded its licensees from developing any competitive software for a period of ninety-nine years.[268] The court held that although the copyright holder undoubtedly had the right to protect against copying of the Interact code, the underlying restrictive licensing agreement went much further and was used in a manner adverse to the public policy embodied in the grant of a copyright.[269]

Copyright misuse is distinguished from an affirmative antitrust violation even though some principles of misuse doctrine overlap antitrust law. As explored above, conduct underlying a copyright misuse defense in an infringement case may serve as a basis for antitrust liability.[270] The misuse defense, however, does not require proof of the additional elements—such as market power, competitive injury, intent to monopolize, etc.—which are necessary to establish an antitrust liability.[271] Meanwhile, proof of an antitrust violation can often serve to establish copyright misuse.[272]

When a contract or conduct involves restraint of trade whose economic function is to restrict, limit, or affect the economic freedom of other parties' actions, such restriction can give rise to an antitrust violation.[273] Yet, a defendant in an infringement action is shielded from suit if he can show copyright misuse, even though the acts of misuse

---

F.2d 832, 846 (Fed. Cir. 1992) ("[T]he Ninth Circuit suggests that, under the appropriate factual setting, copyright misuse may be a viable defense against a claim of copyright infringement."); Data Gen. Corp. v. Grumann Sys. Corp., 36 F.3d 1147, 1170-71 (1st Cir. 1994); DSC Communications Corp. v. DGI Techs., Inc., 81 F.3d 597, 601 (5th Cir. 1996) (holding that "DGI may well prevail on the defense of copyright misuse, because DSC seems to be attempting to use its copyright to obtain a patent-like monopoly over unpatented microprocessor cards").

   267.   911 F.2d 970 (4th Cir. 1990).
   268.   *See id.* at 973.
   269.   *See id.* at 978.
   270.   *See Data General*, 36 F.3d at 1147; United States v. Microsoft, 1998 WL 614485, at *15 (D.D.C. Sept. 14, 1998) ("[C]opyright law does not give Microsoft blanket authority to license (or refuse to license) its intellectual property as it sees fit.").
   271.   *See Lasercomb*, 911 F.2d at 978 ("The question [in a misuse defense] is not whether the copyright is being used in a manner violative of antitrust law . . . but whether the copyright is being used in a manner violative of the public policy embodied in the grant of a copyright."); Zenith Radio Corp. v. Hazeltine Research, Inc., 395 U.S. 100, 140 (1969); Laitram Corp. v. King Crab, Inc., 245 F. Supp. 1019, 1020 (D. Alaska 1965).
   272.   *See* Lande & Sobin, *supra* note 18, at 250.
   273.   *See* Peter C. Carstensen, Concepts for Restraint Analysis: Naked and Ancillary, Internal and External, Dependent and Independent 1 (Sept. 1999) (unpublished manuscript, on file with author).

neither constitute competitive injury nor indicate that the plaintiff was individually harmed by the defendant's misuse. [274] An antitrust violation is a counterclaim giving rise to damages; whereas, misuse is an absolute defense against an allegation of copyright infringement.[275]

Just as importantly, this Article recommends that a contractual license purporting to prohibit or limit software reverse engineering be subject to copyright misuse or antitrust liability. In *Alcatel U.S.A., Inc. v. DGI Technologies, Inc.*,[276] the United States Court of Appeals for the Fifth Circuit ruled that a license agreement prohibiting a licensee from reverse engineering constituted copyright misuse.[277]   If a restrictive license, for the purpose of extending copyright protection beyond its proper scope, prevents competitors from detecting or understanding unprotected ideas or functional elements embodied in the computer program, and from developing competing or interoperable products, it constitutes anticompetitive use of a copyright. The following Parts will explore several types of contractual licenses under UCITA such as "shrink-wrap" or "click-wrap" licenses, which prohibit software reverse engineering, and analyze them from a variety of antitrust angles in great detail.

IV. CONTRACTUAL RESTRICTIONS ON REVERSE ENGINEERING

A. *"Shrink-Wrap" Licenses Under UCITA*

A "shrink-wrap" license prohibiting reverse engineering provides protection over that available under the Copyright Act.[278] A software vendor almost always imposes restrictions against reverse engineering through contractual provisions, the most obvious and traditional means to bind a user.   Off-the-shelf software in mass-market software sales

---

274.   *See* Marshall Leaffer, *Engineering Competitive Policy and Copyright Misuse*, 19 U. DAYTON L. REV. 1087, 1099 (1994); *Zenith Radio Corp.*, 395 U.S. at 1114; Laitram Corp. v. King Crab, Inc., 245 F. Supp. 1019, 1020 (D. Md. 1966).

275.   Morton Salt Co. v. G.S. Suppiger, 314 U.S. 488 (1988); *Laitram*, 245 F. Supp. at 1020.

276.   166 F.3d 772 (5th Cir. 1999). DGI reverse engineered an Alcatel microprocessor card to discover its functional components and to develop its own version of the card. *Id.* at 779.

277.   *Id.* at 793.

278.   A shrink-wrap license agreement is frequently present on an envelope containing the software inside a sealed outer box. It states that if the buyer does not consent to this licensing agreement, he or she should return the unopened software to the place of purchase. Unfortunately, most software retailers refuse to take back software once the outer wrapping is opened. *See* 1 NIMMER & NIMMER, *supra* note 67, § 27.02[B], 27-14. A "click-wrap" license is the electronic equivalent of shrink-wrap licenses. This type of license appears on a user's computer screen when the user first loads a computer program, and requires the user to "click" his acceptance of the terms of the license before he uses the software.

includes boilerplate provisions that contain a prohibition against reverse engineering. These prevent end users who remove the plastic shrink-wrap enclosing the software from copying, decompiling, disassembling or modifying software in an attempt to understand how the program works.

Such shrink-wrap licenses are enforceable under UCITA. A vendor can enforce a contractual prohibition against reverse engineering and restrict interoperability under UCITA. However, as discussed, the copyright balance is achieved through the limitation on the rights of a copyright holder, which is reflected by the idea/expression dichotomy and fair use doctrine or misuse doctrine that provides legitimacy for using certain reverse engineering rights. Even under state trade secret law, it is not a misappropriation to discover or appropriate a trade secret by reverse engineering.[279] It is, therefore, worth discussing in some depth whether shrink-wrap licenses containing prohibitions against reverse engineering of the publicly distributed object code are enforceable. This Part outlines the general problems of UCITA and then discusses the federal law preemption of state law. In following Parts, the doctrine of unconscionability is discussed as an assertion of unenforceability as to shrink-wrap licenses prohibiting reverse engineering.

UCITA began as a proposal to change article 2B of the Uniform Commercial Code (U.C.C.), which dealt with the sale of goods. "Courts treat sales of packaged software as a sale of goods and apply Article 2 of the U.C.C. (Law of Sales) to disputes involving packaged software."[280] Since the development and sale of custom software is not covered by the U.C.C., it is being revised to include a new article 2B (Law of Licensing

---

279. *See* Cataphote Corp. v. Hudson, 422 F.2d 1290, 1293 (5th Cir. 1970); RESTATEMENT (THIRD) OF UNFAIR COMPETITION § 43 (1995) ("Independent discovery and analysis of publicly available products or information are not improper means of acquisition."); Anthony J. Mahajan, *Intellectual Property, Contracts, and Reverse Engineering After ProCD: A Proposed Compromise for Computer Software*, 67 FORDHAM L. REV. 3297, 3318-19 (1999) (explaining that to protect trade secrets for computer source code from reverse engineering, security measures such as password access or encryption beyond simple distribution are necessary).

280. Cem Kaner, *Restricting Competition in the Software Industry—The Impact of Pending Revisions to the U.C.C.*, 3 CYBERSPACE LAWYER 11 (1998); *see also* Advent Sys. v. Unisys Corp., 925 F.2d 670, 675 (3d Cir. 1991) (holding that a computer program may be copyrightable as intangible intellectual property, but once it is copied onto a floppy disc or other medium, it becomes a tangible, moveable, and physical good); RRX Indus., Inc. v. Lab-Con, Inc., 772 F.2d 543 (9th Cir. 1985) (stating that computer software system contract requiring repair of "bugs" was a contract for goods); Triangle Underwriters, Inc. v. Honeywell, Inc., 604 F.2d 737 (2d Cir. 1979), *aff'd*, 651 F.2d 132 (2d Cir. 1981) (stating that sale of computer package including hardware, operating system software and custom application software deemed a contract for the sale of goods).

of Information).[281] Although there is great benefit in creating a uniform legal system for software products and services, this particular proposal for unifying the law is seriously flawed.[282] Part of the difficulty in developing pertinent rules for this type of commerce has been the need to move away from a contract law regime focused on the sale or lease of goods and into a paradigm focused on computer information as the subject matter of transactions. Thus, the National Conference of Commissioners on Uniform State Laws (NCCUL) and the American Law Institute, the two co-sponsors of the U.C.C., decided that article 2B should not remain part of the U.C.C. NCCUL decided to complete and promote the work as a stand-alone uniform act. Removing this project from the U.C.C. reduced the need to reconcile the sale of goods principles with the entirely different matter of transactions in computer information. The new uniform law, known as UCITA, was proposed in July 1999. It provided a critically important framework for state law in this subject area of licensing contracts.[283]

      UCITA has faced criticism because, at the outset, UCITA over-represented one group but under-represented another; many software vendors primarily led the development of the UCITA draft without including consumer groups and other trade organizations. Since the publisher has a concentrated and high-stakes interest as compared to consumers or other software developers, the political process is subject to a distorting minoritarian bias.[284] Indeed, UCITA is very likely to cause considerable disputes over its interpretation because of a series of default rules for software licensing transactions.[285] Even though the freedom-of-contract philosophy remains one of the basic tenets of contract laws, the UCITA's presupposed reliance on such a classical conception will sustain

---

      281.   *See* U.C.C. 2B-105, Reporter's Note 3 (Apr. 15, 1998 Draft).

      282.   *See* Kaner, *supra* note 280, at 11.

      283.   *See* Fred H. Miller & Carlyle C. Ring, *Article 2B's New Uniform: A Free-Standing Computer Information Transactions Act*, *available at* http://www.2bguide.com/docs/nuaa.html (last visited Dec. 9, 2001). Maryland and Virginia were the only states to approve the law in 2000. *Id.*

      284.   *See* NEIL K. KOMESAR, IMPERFECT ALTERNATIVES: CHOOSING INSTITUTIONS IN LAW, ECONOMICS AND PUBLIC POLICY 56, 76, 173, 192 (1994). Four anti-UCITA states—Iowa, North Carolina, West Virginia, and Vermont—adopted "bomb-shelter" legislation in order to prevent a vendor from applying, for instance, Maryland's UCITA law provisions on residents in a bomb-shelter state. *See Mass. Could Be Fifth State to Adopt Anti-UCITA Law*, INFOWORLD (June 4, 2003), *at* http://www.infoworld.com/article/03/06/04/Hnucita_1.html.

      285.   UCITA 2002 Revision, http://www.nccusl.org/nccusl/ucita/UCITA_082602_MEMO_ and_CHART.pdf (last visited Sept. 25, 2002).

a status quo bias, and may not always be suited to the new practicalities of the information economy and modern pluralistic society.[286]

    In particular, UCITA would not only make the terms of shrink-wrapped licenses more enforceable, but also outlaw reverse engineering to fix problems with software or to make products that work with other products.[287] When copies of software are sold, the intellectual property doctrine of "fair use" protects this user freedom. However, UCITA throws in doubt whether licensing of the same software permits the licensor to restrict fair use by contract.[288]

    The NCCUSL has recently attempted to revise several controversial provisions.[289] The NCCUSL committee has recommended that instead of banning reverse engineering, new language allows reverse engineering for system interoperability.[290] Although the NCCUSL is recommending several changes to amend the problems contained in UCITA, it is still unclear whether these changes will go far enough to help consumers or small business customers. Copyright owners can still forbid software reverse engineering for purposes of testing for security holes or achieving comparability due to a licensor's unseen shrink terms.[291]

    A shrink-wrap license prohibition against reverse engineering generally performs in mass markets. A mass-market transaction is a consumer transaction and any other transaction for information or informational rights directed to the general public as a whole under substantially the same terms for the same information with an end-user license.[292] The mass-market contracts are not contracts between a few individuals, but rather for the general public.[293] Because UCITA does not recognize the sale of a copy of a software product as a sale, "the buyer

---

    286. *See* Clarke, *supra* note 11, at 6-7; Robert A. Hillman, *Debunking Some Myths About Unconscionability: A New Framework for U.C.C. Section 2-302*, 67 CORNELL L. REV. 1, 41, 43 (1981); Richard A. Epstein, *Unconscionability: A Critical Reappraisal*, 18 J. L. & ECON. 293, 294-95 (1975).

    287. As other issues, it would give vendors the right to repossess software by disabling it remotely; prevent the transfer of licenses from one party to another without vendor permission; and allow vendors to disclaim warranties.

    288. *See* Jean Braucher, *Why UCITA, Like U.C.C. Article 2B, Is Premature and Unsound*, 1 U.C.C. BULL 1, July 1999.

    289. The proposed changes: electronic self-help banned; a state's consumer protection law trumps UCITA; right to criticize protected; remedies for known material defect preserved; reverse engineering for interoperability expressly authorized; special open-source software provisions. *See id.*

    290. *See* Amendments to Uniform Computer Information Transaction Act, Amendment #6: New Section 118 (Meeting in its One-Hundred-and-Eleventh Year, June 26-August 2, 2002), http://www.law.upenn.edu/bll/ulc/ulc_frame.htm (last visited Sept. 25, 2002).

    291. *See* Ed Foster, *Bride of UCITAstein*, INFOWORLD, Jan 14, 2002, at 64.

    292. UCITA § 102(44)-(45) (2002).

    293. *Id.*

becomes a licensee and not an owner of a copy, and thus the first sale doctrine of copyright is never triggered."[294] In every other industry in the United States, mass market distribution of a product carries with it a grant to the customer of the right to reverse engineer the product. Therefore, it is questionable what benefit there is to the public from a law that lets software owners limit the research opportunities of their competitors in ways not allowed for other industries.

     A shrink-wrap contract in mass-market transactions is a nonnegotiated standard form contract. With extremely few exceptions, all of the terms in this "license agreement" will be fully enforceable against the customer, as if he/she had reviewed, discussed, and signed a paper contract before the sale.[295] Such a nonnegotiable standard contract, however, may be unenforceable under the doctrine of unconscionability and federal preemption theory, another important topic this Article will discuss.

B.   Federal Preemption vs. Antireverse Engineering Contracts

1.   UCITA Sections 105(a)-(b)

     Not every contract term prohibiting reverse engineering can be enforced under UCITA. If an antireverse engineering term is contrary to fundamental public interests, courts could refuse to enforce the term. Section 105 of UCITA suggests that contractual terms should be consistent with modern social policy. Despite this concern with public interests, the underlying section seems to undermine the balance between licensors and users by granting overriding protection for boilerplate provisions that purports to limit users' rights such as reverse engineering, publishing results of testing, and other criticisms. Because of uncertainty in relation to fundamental public policy, a contractual device such as a shrink-wrap license is used to waive federally created privilege to make fair use of copyrighted works.[296] The contract is a subject of state law. Since the transformations caused by digital information, state common and statutory law have often clashed with federal laws in areas of patents and copyrights.   The issue of whether an antireverse engineering provision that limits the fair use right protected under copyright law is enforceable is one of the extensive conflicts.

---

     294.   Kaner, *supra* note 280, at 14.
     295.   *See* BAND & KATOH, *supra* note 8, at 220-21.
     296.   *See* Charles R. McManis, *The Privatization (Or "Shrink-Wrapping") of American Copyright Law*, 87 CAL. L. REV. 173, 175 (1999).

UCITA provides courts with an instructive guideline on how the issue of preemption is solved even though it does not address the issue. Section 105(a) states that "[a] provision of this [Act] which is preempted by federal law is unenforceable to the extent of the preemption."[297] A particular state law rule or contract term may be invalidated because of federal preemption.[298] A contract term that varies the effect of federal rule that cannot be varied by agreement under the Copyright Act is unenforceable. Subsection (a) refers to preemption, but other doctrines grounded in federal law may preclude enforcement of some contract terms in some cases.[299] For example, a finding of IP misuse or the violation of antitrust law could support federal law preemption with respect to federal policies.[300] UCITA, however, does not define when federal preemption may occur. Accordingly, an antireverse engineering provision under a licensing contract may be unenforceable if it denies federally created privilege, the fair use doctrine.

Section 105(b) sets out the fundamental public policy principle of UCITA. Subsection (b) provides:

> If a term of a contract violates a fundamental public policy, the court may refuse to enforce the contract, enforce the remainder of the contract without the impermissible term, or limit the application of the impermissible term so as to avoid a result contrary to public policy, in each case to the extent that the interest in enforcement is clearly outweighed by a public policy against enforcement of the term.[301]

Thus, contract terms may be unenforceable if they violate a fundamental public policy. However, public policy should "clearly override" policy favoring enforcement of private transactions between the parties. The Official Comment suggests that "[i]n the absence of a legislative declaration of a particular policy, courts should be reluctant to override a contract term."[302] Uncertainty embedded in the statutory language creates difficult burdens for users to meet in litigation and would thereby chill exercise of user's freedoms and cultural development.[303]

---

297. UCITA § 105(a) (2002).
298. *Id.* § 105 cmt. 2.
299. *Id.*
300. *See* Maureen A. O'Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms*, 45 DUKE L.J. 479, 551 (1995) (arguing that antidecompilation provisions should be preempted if in violation of federal antitrust laws).
301. UCITA § 105(b).
302. *Id.* § 105 cmt. 3.
303. Letter from Cem Kaner, a working group within the Business Law Section of the American Bar Association, to president Gene Lebrun and other commissioners of NCCUSL

The offsetting public policies are mostly related to innovation, competition, fair comment and fair use.[304] "Innovation policy recognizes the need for a balance between protecting property interests in information to encourage its creation and the importance of a rich public domain upon which most innovation ultimately depends. Competition policy prevents unreasonable restraints on publicly available information in order to protect competition."[305] Although the fair use doctrine is established by Congress in the Copyright Act, the policies established on fair use extend to section 105 for consideration and deliberation.[306]

Although it does not address the issue of national policy on reverse engineering, subsection (b) recognizes, at least, a "policy not to prohibit some reverse engineering where it is needed to obtain interoperability of computer programs."[307]

This Article recommends that courts should consider the public policies underlying the fair use doctrine in determining if a provision prohibiting reverse engineering is enforceable. Although the Congress grants a copyright holder monopoly rights for his/her work, this does not extend to matters that are subject to independent discovery, fair use, and the merger doctrine.[308] Therefore, if intermediately copying a process provides legitimacy under the fair use tests, courts should override an antireverse engineering provision.

The third Official Comment to § 105 provides various factors that courts should consider in evaluating a claim that a term violates fundamental public policy:

> [1] the extent to which enforcement or invalidation of the term will adversely affect the interests of each party to the transaction or the public, [2] the interest in protecting expectations arising from the contract, [3] the purpose of the challenged term, [4] the extent to which enforcement or invalidation will adversely affect other fundamental public interests, [5] the strength and consistency of judicial decisions applying similar policies in similar contexts, [6] the nature of any express legislative or regulatory

---

(June 10, 1999), *at* http://legalminds.lp.findlaw.com/list/rre/frm00246.html (last visited Feb. 10, 2003).

    304. UCITA § 105 cmt. 3.

    305. *Id.*

    306. *Id.*

    307. *Id.* (citing 17 U.S.C. § 1201 (2000)).

    308. *See* Eric Douma, *The Uniform Computer Information Transaction Act and the Issue of Preemption of Contractual Provisions Prohibiting Reverse Engineering, Disassembly, or Decompilation*, 11 ALB. L.J. SCI. & TECH. 249, 271-72 (2001).

policies, and [7] the values of certainty of enforcement and uniformity in interpreting contractual provisions.[309]

The third Official Comment to § 105 also recognizes that "[c]ontracting parties may have greater freedom contractually to restrict the use of confidential information than information that is otherwise publicly available."[310] While it agrees that "[t]rade secret law allows information to be transferred subject to considerable contractual limitations on disclosure which facilitates the exploitation and commercial application of new technology," it also stresses that "trade secret law does not prohibit reverse engineering of lawfully acquired goods available on the open market [and] [s]triking the appropriate balance depends on a variety of contextual factors that can only be assessed on a case-by-case basis with an eye to national policies."[311]

Section 105(a) and (b) propose to "strike the balance between fundamental interests in contract freedom and fundamental public policies such as those regarding innovation, competition, and free expression."[312] The instrumental idea of contract law is based on free-contracting philosophy in which parties are free to choose terms. However, a shrink-wrap license prohibiting reverse engineering in mass market is not based on parties' negotiation, but rather on a nonnegotiable form contract. Indeed, UCITA does not alter intellectual property or other relevant fundamental information laws such as the Constitution, competition or trade regulation law.[313] For example, it assures the right to access to information for public purposes, such as education, research, and fair comment, and that the information in the public domain is free for all to use.

2.   Copyright and Fair Use

Section 301(a) of the Copyright Act is intended to preempt and abrogate any rights under the common law or statutes of any state, which are equivalent to the exclusive rights federally created by Copyright.[314] To

---

309.   UCITA § 105 cmt. 3.
310.   *Id.*
311.   *Id.*
312.   *Id.* § 105 cmt. 1.
313.   *Id.* The enforceability of the license terms is determined under this Act and other applicable law, including copyright law. *See id.* § 102 cmt. 37.
314.

[A]ll legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106 in works of authorship that are fixed in a tangible medium of expression and come within the subject matter of copyright as specified by sections 102 and 103, whether created before or after that

establish preemption, § 301 sets a two-step analysis. The first step requires that the work of authorship, in which rights are claimed under the state statute, must fall within the subject matter of copyright as specified by §§ 102 and 103.[315] The second step requires the state law to create legal or equitable rights that are equivalent to the rights created by the Copyright Act.[316] Therefore, if a state law right is within the general scope of copyright, preemption would occur. Regarding a contractual restriction on software reverse engineering, computer programs constitute copyrightable subject matter as literary works under § 102(a)(1).[317] Then, the question is whether a contractual limitation on reverse engineering would create legal or equitable rights equivalent to rights specified by § 106 of the Copyright Act.

     Since the federal statute fails to define equivalency, each court has to distinguish for itself between equivalent rights and nonequivalent rights.[318] Courts have analyzed this preemption inquiry by applying the "extra element" test.[319] This test asks whether a state law right requires proof of an extra element that is qualitatively distinctive from the federal right.[320] Preemption may be denied if one or more qualitatively different elements, which constitute the state-created cause of action, are found in addition to those required for copyright infringement.[321] However, some criticism of this test indicates that this decision is arbitrarily made by courts rather than on the basis of any real guidance; "[t]here is always

---

     date and whether published or unpublished, are governed exclusively by this title. Thereafter, no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State.

17 U.S.C.A. § 301(a) (2000).

     315.   *See* Harper & Row Publishers, Inc. v. Nation Enters., 773 F.2d 195, 200 (2d Cir. 1983), *rev'd on other grounds*, 471 U.S. 539 (1985).

     316.   *Id.*

     317.   17 U.S.C. § 102(a)(1) (2000).

     318.   *See generally* Patrick McNamara, *Copyright Preemption: Effecting the Analysis Prescribed by Section 301*, 24 B.C.L. Rᴇᴠ. 963, 966-68 (1983) (explaining that 17 U.S.C. § 101 lists no explanation of the term "equivalency").

     319.   *See* Alcatel U.S.A, Inc. v. DGI Techs. Inc., 166 F.3d 772, 787 (5th Cir. 1999); Trandes Corp. v. Guy F. Atkinson Co., 996 F.2d 655, 659 (4th Cir. 1993); Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 719 (2d Cir. 1992); Harper & Row Publishers, Inc. v. Nation Enters., 501 F. Supp. 848, 852 (S.D.N.Y. 1980), *aff'd*, 723 F.2d 195 (2d Cir. 1983), *rev'd on other grounds*, 471 U.S. 539 (1985); 1 Nɪᴍᴍᴇʀ & Nɪᴍᴍᴇʀ, *supra* note 67, § 1.01[B][1], 13.

     320.   *See Alcatel*, 166 F.3d at 787.

     321.   *See* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1454 (7th Cir. 1996) (holding that mutual assent and consideration required by a contract claim render that claim qualitatively different from copyright infringement.); Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 716 (2d Cir. 1992) ("[I]f an 'extra element' is 'required instead of or in addition to the acts of reproduction, performance, distribution or display, in order to constitute a state-created cause of action, then the right does not lie "within the general scope of copyright," and there is no preemption.'" (citations omitted)).

some difference between the state law and the Copyright Act."[322]   If a court wants to avoid preemption, it can always find some difference that becomes the "extra element" needed to avoid preemption.[323]   In contrast, when there is preemption, no "extra element" is put on the label.[324]

This Article argues that a right created by contractual provisions against reverse engineering would be equivalent to rights granted under the Copyright Act. The issue is resolved by an analysis of the legislative intent rather than simply finding out an extra element such as existence of a valid contract. Since Congress has the power to preempt state law in a given area, the matter of its intent is whether Congress has in fact exercised such power.[325]   The legislative history indicates that the primary purpose of § 301 was to preempt the common-law copyright protection for unpublished works, which coexisted with the federal statutory copyright protection for published works prior to the enactment of the 1976 Copyright Act.[326]   It also states that "[a]s long as a work fits within one of the general subject matter categories of sections 102 and 103, the bill prevents the States from protecting it even if it fails to achieve Federal statutory copyright [protection]."[327]   This suggests that § 301(a) itself might preempt any effort to provide contractual protection against reverse engineering of a publicly distributed computer program, where the program would fail to achieve federal copyright protection against such reverse engineering because of the fair use provision of § 107.[328]

The intent of Congress regarding preemption also appears in relation to § 301(b) of which the purpose "is to make clear, consistent with the 1964 Supreme Court decisions in [the] Sears . . . and Compco [cases] . . . that preemption does not extend to causes of action, or subject matter outside the scope of the revised Federal copyright statute."[329]   "This statement provides authority for applying the *Sears-Compco* preemption test to state shrink-wrap licensing laws and other contractual

---

322.   Schuyler Moore, *Straightening Out Copyright Preemption*, 9 UCLA ENT. L. REV. 201, 204 (2002).

323.   *See id.*

324.   *Id.*

325.   *See* Pac. Gas & Elec. Co. v. State Energy Comm'n, 461 U.S. 190, 203 (1983); Stephen A. Gardbaum, *The Nature of Preemption*, 79 CORNELL L. REV. 767 (1994).

326.   *See* H.R. REP. NO. 94-1476, 94th Cong., 2d Sess. 129-131, *reprinted in* 5 U.S.C.C.A.N. 5745-47 (1976).

327.   *Id.* at 5747.

328.   *See* Charles R. McManis, *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States and the European Community*, 8 HIGH TECH. L.J. 25, 89 (1993).

329.   H.R. REP. NO. 94-1476, 94th Cong., 2d Sess. 129-31, *reprinted in* 5 U.S.C.C.A.N. 5745, 5747 (1976).

restrictions on reverse engineering, even if § 301 itself does not preempt such provisions."[330]

A contractual provision restricting the entire public or a particular party from reverse engineering might create protection that is equivalent to § 106 rights because the provision prohibits copying. Section 106 rights are limited by the fair use doctrine; any limitation on fair use rights broadens the § 106 rights of the copyright holder against public policy.[331] In other words, a contractual restriction on reverse engineering expressly narrows the scope of fair use rights, which have been judicially drawn and allowed in § 107. Thus, an action to enforce a contractual limitation on fair use rights results in the equivalent effects to an infringement claim under Copyright Act.[332]

In *Wright v. Warner Books*, the United States Court of Appeals for the Second Circuit addressed that preemptive effect of § 107 with respect to contractual restrictions on the right to make use of unpublished letters.[333] The court held that "[t]o read [the restrictions agreed upon] as absolutely forbidding any quotation, no matter how limited or appropriate, would severely inhibit proper, lawful scholarly use and place an arbitrary power in the hands of the copyright owner going far beyond the protection provided by law."[334] The Second Circuit's holding strongly suggests that the court would preempt enforcement of a contract attempting to restrict the fair use privilege.[335] In *Symantec Corp. v. McAfee Associates, Inc.*, the court held that a software license prohibiting reverse engineering was preempted by § 301.[336]

3.   The Supremacy Clause

The Supremacy Clause of the Constitution provides a more feasible ground for preemption questions arising out of the application of § 301 of the Copyright Act.[337] In determining whether a state statute is pre-empted by federal law and therefore invalid under the Supremacy Clause of the Constitution, the test also relies on ascertaining the intent of

---

330.   MCMANIS, *supra* note 328, at 90.
331.   Douma, *supra* note 308, at 259.
332.   *See id.*; *see also* BAND & KATOH, *supra* note 8, at 221.
333.   953 F.2d 731, 736-41 (2d Cir. 1991).
334.   *Id.* at 741 (quoting Salinger v. Random House, Inc., GSO F. Supp. 413, 427 (S.D.N.Y. 1986).
335.   *See* MCMANIS, *supra* note 328, at 90.
336.   1998 WL 740798, at *4-*5 (N.D. Cal. June 9, 1998).
337.   The Supremacy Clause provides that the "Constitution, and the Laws of the United States which shall be made in Pursuance thereof . . . shall be the supreme Law of the Land; and Judges in every State shall be bound thereby, and Thing in the Constitution or Laws of any State to the Contrary notwithstanding." U.S. CONST. art. VI, § 1, cl. 2.

Congress. Federal law may supersede state law in several different ways: first, Congress has mandated by express decree; second, "the scheme of federal regulation is sufficiently comprehensive to make reasonable the inference that Congress 'left no room' for supplementary state regulation;" and third, a state law conflicts with federal law when "the state law stands 'as an obstacle to the accomplishment and execution of the full purposes and objectives of Congress.'"[338]

Congress grants certain exclusive rights or monopoly privileges to creators, but at the very same time, courts and legislatures have seen the need to limit the scope of copyright monopoly to strike a copyright balance.[339] The copyright principles and doctrines such as the fair use, first sale doctrine, the idea/expression dichotomy, and the copyright misuse doctrine reflect this need for limitations on copyrights. Theses limitations are an important step toward achieving a uniquely constitutional goal: promoting the progress of science and the useful arts.[340] As one of the "built-in accommodations" contained in copyright law, the fair use doctrine furthers this constitutional goal, which "allows authors and others to bring to market expressive works of potentially great public benefit that are made possible only through the 'fair use' of another's expressive work."[341]

While the right of software users to reverse engineer is not established by statute, no federal court of appeals has disagreed with the notion that reverse engineering constitutes fair use in certain circumstances.[342] Fair use activities often could result in the creation of new and useful software products. Clearly, in light of the constitutional goal of promoting the progress of science and the useful arts through the balance established in the Copyright Act, a prohibition of such activity would stand as an obstacle to the full purpose of Congress. Thus, under the Supremacy Clause analysis, such restrictions on reverse engineering should be preempted.

---

338. Cal. Fed. Sav. & Loan Ass'n v. Guerra, 479 U.S. 272, 280-81 (1987) (citations omitted).

339. *See* Son, *supra* note 264, at 114.

340. U.S. CONST. art. I, § 8, cl. 8; *see* Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 575 (1994).

341. Garry L. Founds, *Shrinkwrap and Clickwrap Agreements: 2B Or Not 2B?*, 52 FED. COMM. L.J. 99, 115 (1999); *see* Eldred v. Ashcroft, 123 S. Ct. 769, 774 (2003).

342. *See* Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000); Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1539 n.18 (11th Cir. 1996); Atari Games Corp. v. Nintendo of Am., 975 F.2d 832, 842-43 (Fed. Cir. 1992); Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1527-28 (9th Cir. 1992), as amended, 1993 U.S. App. LEXIS 78 (9th Cir. Jan. 6, 1993); Mark A. Lemley, *Intellectual Property and Shrinkwrap Licenses*, 68 S. CAL. L. REV. 1239, 1247 n.31 (1995).

In *Vault Corp. v. Quaid Software, Ltd.*,[343] the United States Court of Appeals for the Fifth Circuit applied a Supremacy Clause analysis to set aside a contractual restriction on reverse engineering, noting that the provision "conflicts with the rights of computer program owners under § 117 and clearly 'touches upon an area' of federal copyright law."[344] The court then found that a Louisiana statute permitting enforcement of shrink-wrap licenses was preempted under federal copyright law.[345] Likewise, in *Associated Film Distribution Corp. v. Thornburg*, the Federal District Court held that a state statute regulating the licensing of motion pictures was preempted by copyright law, stating that the "more general question of conflict of the two statutory schemes under the Supremacy Clause is decisive."[346]

The Supremacy Clause also provides the vitality of federal patent law preemption of state law. In *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*,[347] the United States Supreme Court held unanimously that a Florida statute prohibiting duplication of unpatented boat hulls using a direct molding process was preempted by the Supremacy Clause of the Constitution; it conflicted with the strong federal policy of the patent law.[348] The *Bonito Boats* decision reached back to reaffirm the view of the two landmark decisions, *Sears, Roebuck & Co. v. Stiffel Co.*[349] and *Compco Corp. v. Day-Brite Lighting, Inc.*,[350] which hold that copying of the article itself that is unprotected by the federal patent and copyright laws cannot be forbidden by state law.[351] The Court stated that "[b]y offering patent-like protection for ideas deemed unprotected under the present federal scheme, the Florida statute conflicts with the 'strong federal policy favoring free competition in ideas which do not merit patent protection.'"[352] It further stated that the competitive reality of reverse engineering may act as a spur to the inventor, creating an incentive to develop inventions.[353] Since the Florida statute forbids the entire public from engaging in a form of reverse engineering of a product

---

343.   847 F.2d 255 (5th Cir. 1988).
344.   *Id.* at 270.
345.   *Id.* at 269-70.
346.   520 F. Supp. 971, 973 (E.D. Pa. 1981); *rev'd and remanded on other grounds*, 683 F.2d 808 (3d Cir.), *cert. denied*, 480 U.S. 933 (1982).
347.   489 U.S. 141 (1989).
348.   *Id.* at 168.
349.   376 U.S. 225 (1964).
350.   376 U.S. 234 (1964).
351.   *Bonito Boats*, 489 U.S. at 165.
352.   *Id.* at 168 (citing Lear, Inc. v. Adkins, 395 U.S. 653, 656 (1969)).
353.   *See id.* at 160.

publicly available, it substantially reduces the competitive incentive and erodes the general rule of free competition.[354]

C.   *Doctrine of Unconscionability*

Contract terms prohibiting reverse engineering may be unenforceable because they are unconscionable. UCITA adopts the doctrine of unconscionability from article 2 of the U.C.C., enabling courts to police against contract terms which they find to be unconscionable.[355] Section 111 of UCITA provides:

> If a court as a matter of law finds a contract or a term thereof to have been unconscionable at the time it was made, the court may refuse to enforce the contract, enforce the remainder of the contract without the unconscionable term, or limit the application of the unconscionable term so as to avoid an unconscionable result.[356]

The principle of this section is to prevent oppression and unfair surprise, and not to disturb allocation of risks, because of superior bargaining power.[357] The basic test is whether contracts or clauses involved are so one-sided and unfair as to be unconscionable under the circumstances existing at the time the contract was made.[358] This statutory language provides courts with uncertainty and ambiguity in judicial determination of what kinds of contracts or bargaining procedure could be considered unconscionability. It also does not clarify the scope of what constitutes oppression or unfair surprise.

In general, unconscionability cases involving bargaining misconduct can be decided under the standard contract-law defenses, such as fraud, duress, undue influence, mistake, inability, take-it-or-leave-it negotiating postures, or illegality.[359] As for determination of unconscionability cases, Professor Leff suggests a noteworthy distinction between bargaining naughtiness as "procedural unconscionability" and gross overall imbalance of an entire contract as "substantive unconscionability."[360] According to this distinction, a shrink-wrap license

---

354. *See id.* at 161.

355. *See* UCITA § 105 & § 111 (2002); U.C.C. § 2-302 (1998 Official Text).

356. UCITA § 111.

357. *See id.* § 111 cmt. 2; Intel Corp. v. Intergraph Corp., 195 F.3d 1346, 1365 (Fed. Cir. 1999)).

358. *See* UCITA § 111 cmt. 2; MIMEO 1941 DRAFT § 1-C, cmt. A(3), at 18 (explaining that in the Sales field, the terms are entirely lopsided, but are fair because they correct an inappropriate condition in a reasonable way).

359. *See* RESTATEMENT (SECOND) OF CONTRACTS § 19 (1981).

360. Arthur Allen Leff, *Unconscionability and the Code—The Emperor's New Clause*, 115 U. PA. L. REV. 485, 487, 509 (1967).

prohibiting reverse engineering in mass market transactions might be located within the realm of procedural unconscionability.[361]   The contracting procedure element which will permit scrutiny for unconscionability is not the mere use of a form contract but the use of a form plus some "vice."[362]   The procedure element has identified this form-plus situation with the "contract of adhesion":[363] a contract to which one of the parties must either adhere entirely or refuse altogether with some of the powers of a monopolist.[364]   Contracts of adhesion are offered on a nonnegotiable take-it-or-leave-it basis by a party having superior bargaining position.

     A shrink-wrap license restricting reverse engineering may be void if imposed in a take-it-or-leave-it option, but will be enforced if embodied in an agreement reflecting deliberative assent in a commercial setting. In *Vault Corp. v. Quaid Software, Ltd.*, the court stated that a shrink-wrap license is unenforceable as a "contract of adhesion."[365] A licensor of the adhesion contract in the position to refuse to bargain for some reason gave a licensee a take-it-or-leave-it option.

     Likewise, the United States Court of Appeals for the Third Circuit in *Step-Saver Data Systems, Inc. v. Wayse Technology*,[366] invalidated the box-top license under a battle of the forms analysis. The case involved a breach of warranty claim brought by Step-Saver Data Systems, Inc. (Step-Saver), the purchaser of the shrink-wrapped computer software, against vendor, The Software Link, Inc. (TSL).[367] Step-Saver purchased and resold 142 copies of TSL's Multilink Advanced Program.[368] Step-Saver obtained copies of the program by placing telephone orders with TSL.[369] The software was then mailed to Step-Saver, along with a box-top licensing agreement disclaiming all express and implied warranties.[370] The shrink-wrap license stated that "[o]pening this package indicates your acceptance of these terms and conditions."[371]

---

     361.   *See id.* at 504-05.
     362.   *See id.*; 1 WILLIAM HAWKLAND, A TRANSACTIONAL GUIDE TO THE UNIFORM COMMERCIAL CODE § 1.1602 (1964); Elvin R. Latty, *Sales and Title and the Proposed Code*, 16 LAW & CONTEMP. PROB. 3, 19 n.78 (1951).
     363.   Todd D. Rakoff, *Contracts of Adhesion: An Essay in Reconstruction*, 96 HARV. L. REV. 1173 (1983).
     364.   *See* Leff, *supra* note 360, at 505.
     365.   847 F.2d 255, 269 (5th Cir. 1988).
     366.   939 F.2d 91, 97-105 (3d Cir. 1991).
     367.   *Id.*
     368.   *Id.* at 95.
     369.   *Id.*
     370.   *Id.*
     371.   *Id.* at 97.

When copies of the computer program did not function properly, at least twelve of Step-Saver's customers filed suit against Step-Saver for damages.[372] Step-Saver then subsequently brought a breach of warranty action against TSL. Step-Saver argued that the contract was formed on the telephone when TSL agreed to ship the copy at the agreed-upon price.[373] This would make the shrink-wrap license a material alteration to the contract between the parties and would therefore not become part of the contract under U.C.C. § 2-207(2).[374]

TSL contended that formation did not occur until Step-Saver received the program, saw the terms of the license, and opened the packaging.[375] Alternatively, TSL argued that its acceptance of Step-Saver's telephone offer was conditional on Step-Saver's own acceptance of the shrink-wrap license.[376] It also argued that Step-Saver was aware of the warranty disclaimers contained in the shrink-wrap license, and by continuing to order and accept copies of the program with such knowledge, Step-Saver implicitly assented to the disclaimers.[377]

The Third Circuit rejected all of TSL's arguments. The court held that the parties did not mutually intend for the shrink-wrap license to constitute the final expression of, or a binding modification to, the agreement reached by the parties.[378] It concluded that the contract was sufficiently definite without reference to the shrink-wrap license because all of the necessary terms to form a contract were present including the identification of the goods, the quantity, and the price.[379]

As for the test to determine whether a shrink-wrap license constituted a conditional acceptance, the court adopted an approach that required "the offeree to demonstrate an unwillingness to proceed with the transaction unless the additional or different terms are included in the contract."[380] The court held that the shrink-wrap licensing agreement was an insufficient indication that TSL was willing to forego the transaction altogether unless Step-Saver assented to the additional terms of the

---

    372. *Id.* at 94.
    373. *Id.* at 97.
    374. *Id.*
    375. *Id.*
    376. *Id.* at 97-98.
    377. *Id.* at 98.
    378. *Id.* In fact, the president of Step-Saver testified that he objected to the terms of the shrink-wrap license. *Id.*
    379. *Id.* at 100.
    380. *Id.* at 102.

shrink-wrap license.[381] Consequently, it determined that the shrink-wrap provisions were not part of the contract between the parties.[382]

On the other hand, the United States Court of Appeals for the Seventh Circuit in *ProCD, Inc. v. Zeidenberg*[383] held that a shrink-wrap agreement was an enforceable contract that prohibited against copying ProCD's uncopyrightable data.[384] It further ruled that the enforcement of a shrink-wrap license was not preempted by the Copyright Act because the license under state law did not create rights equivalent to the exclusive rights within the general scope of copyright.[385]

ProCD compiled information from more than 3000 telephone directories into a computer database.[386] ProCD sold a version of the database, called "SelectPhone," on CD-ROM discs. SelectPhone is a program that has a copyright.[387] The uncopyrightable database in SelectPhone cost more than $10 million to compile and is expensive to keep current.[388] ProCD decided to engage in price discrimination, selling its database to the general public for personal use at a low price, while selling information to the manufacturers and retailers for a higher price.[389] Every box containing its consumer product states that the software inside comes with restrictions based on an enclosed license.[390] This license, which appears on the user's screen every time the software runs, restricts use of the application program and listings to noncommercial purposes only. Matthew Zeidenberg bought a consumer package of SelectPhone and decided to ignore the license terms.[391] He formed Silken Mountain Web Services, Inc., to resell the information in the SelectPhone database.[392] Sillken made the database available on the Internet for a lower price than what ProCD charges its commercial customers.[393]

The crux of the matter was whether rights created by contract are equivalent to any of the exclusive copyright rights. In this case, the court ruled that federal preemption clauses generally left contract parties

---

381. *Id.* at 103.
382. *Id.*
383. 86 F.3d 1447 (7th Cir. 1996).
384. *Id.* at 1454.
385. *Id.* at 1455.
386. *Id.* at 1449.
387. *Id.* at 1450.
388. *Id.*
389. *Id.* 1449-50.
390. *Id.* at 1450.
391. *Id.* at 1447.
392. *See id.*
393. *Id.*

unaffected.[394]   Unlike copyrights, contracts generally affect only the parties, and contractual terms reflect private ordering that is essential to the efficient functioning of market.[395]   The court held that the federal statute should not generally preempt the enforcement of contractual terms and conditions; for example, § 301(a) should not forbid states from substituting their own regulatory systems for those of the national government.[396]   Finally, the court held that a shrink-wrap license was a simple two-party transaction that was not equivalent to any of the exclusive copyrights.[397]

The Supreme Court in *Feist Publication, Inc. v. Rural Telephone Service Co.* held that purely factual data was not copyrightable.[398]   The Seventh Circuit's decision allows for a software vendor to obtain copyright-like protection (beyond the scope of copyright law) for an electronic database—such as that within the Select Phone database—by contractual restriction.   Professor Netanel, criticizing this contradictory holding, states that "standardized contracts that systematically proscribe user copying of public domain material may frustrate the social policy behind copyright law's delicate balance of incentive and access."[399]

Meanwhile, the Seventh Circuit did not apply the *Feist* Court's copyright analysis to ProCD's copyright claim even though ProCD's electronic database may be not only creative, but also distinctive from Rural's basic telephone white pages.[400]   In that regard, commentators have recently suggested that computer databases be protected under existing copyright law if it is sufficiently creative to satisfy the originality requirement.[401]

In *Bowers v. Baystate Technologies, Inc.*,[402] the Federal Circuit has recently taken one step back in reconciling a federal copyright law and state contract law by indubitably holding that the Copyright Act does not preempt the prohibition of reverse engineering embodied in a shrink-

---

394.   *Id.* at 1454.

395.   *Id.*

396.   *Id.* at 1455; *see* Am. Airlines, Inc. v. Wolens, 513 U.S. 219 (1995); Norfolk & W. Ry. v. Train Dispatchers, 449 U.S. 117 (1991).

397.   *ProCD*, 86 F.3d at 1454.

398.   499 U.S. 340, 364 (1991).

399.   Neil Weinstock Netanel, *Copyright and a Democratic Civil Society*, 106 YALE L.J. 283, 385 (1996).

400.   *See ProCD*, 86 F.3d at 1449.

401.   *See, e.g.*, Katya Cullberg, *Copyright and Compilations: Protecting the Data in Computer Databases*, 6 No. 2 INTELL. PROP. L. BULL. 15 (2001); Brett L. Tolman, Note, *ProCD, Inc. v. Zeidenberg: The End Does Not Justify the Means in Federal Copyright Analysis*, 1998 B.Y.U. L. REV. 303, 318 (1998).

402.   320 F.3d 1317 (Fed. Cir. 2003).

wrap license agreement.[403]   Mr. Bowers created computer aided design (CAD) software and a template to improve CAD software, and bundled a DOS-based add-on program to operate with CAD and Cadjet as the Designer's Toolkit in 1989.[404]   Mr. Bowers marketed the Designer's Toolkit with a shrink-wrap license that prohibited any reverse engineering.[405]

     Baystate also developed and sold tools such as Draft-Pak version 1 and 2 for CADKEY.[406]   In 1991, it obtained copies of Mr. Bowers' Designer's Toolkit and developed the substantially revised Draft-Pak version 3, incorporating many of the features of Designer's Toolkit.[407] The new version of Draft-Pak induced intense price competition between Mr. Bowers and Baystate.[408]   To gain market share over Baystate, Mr. Bowers negotiated with Cadkey, Inc., to provide the Designer's Toolkit free with CADKEY.[409]   However, Cadkey, Inc. refused to enter a distribution agreement with Mr. Bowers because of pressure from Baystate.[410]   Finally, Baystate purchased Cadkey, Inc. and eliminated Mr. Bowers from the CADKEY network so that he was unable to market the Designer's Toolkit for that program.[411]   Mr. Bowers filed counterclaims for copyright infringement, patent infringement, and breach of contract in response to Baystate's suit for declaratory judgment that its products did not infringe on Mr. Bowers' patent and that the patent was invalid.[412]

     In rejecting Baystate's preemption argument, the Federal Circuit held that the Copyright Act did not preempt enforcement of a total ban on reverse engineering under a shrink-wrap license; the court seemed to stand at the far left of the IP spectrum by providing freedom of contract with much respect.[413]   During the process, the court relied heavily on the First Circuit's approach in *Data General Corp. v. Grumman System Support Corp.* that the Copyright Act did not preempt the state law trade secret claim if additional elements of proof such as a trade secret and breach of a duty of confidentiality were qualitatively different from a copyright claim.[414]   Although the *Data General* decision did not expressly

---

     403.   *Id.*
     404.   *Id.* at 1320-21.
     405.   *Id.* at 1322.
     406.   *Id.*
     407.   *Id.*
     408.   *Id.*
     409.   *Id.*
     410.   *Id.*
     411.   *Id.*
     412.   *Id.*
     413.   *See id.* at 1324-26.
     414.   36 F.3d 1147, 1168 (1st Cir. 1994).

address the federal law preemption, the Federal Circuit presumed that the rationale behind *Data General* would lead to a judgment that private parties were free to contractually forego the limited ability to reverse engineer software under the exemptions of the Copyright Act.[415]

Like a UCITA approach, this court's logic portrays a shrink-wrap license as a freely-entered agreement even if a shrink-wrap license overrides the fair use doctrine. In fact, the approach permits state law to eviscerate an important federal copyright policy reflected in the fair use defense,[416] and, indeed, the Federal Circuit rendered a decision in conflict with other federal courts of appeals' decisions.[417]

As discussed above, the test for state law preemption under § 301 of the Copyright Act should be whether the state law "substantially impedes the public use of the otherwise unprotected" material.[418] In dissenting from the majority opinion, Judge Dyk would have held:

> A state law that allowed a copyright holder to simply label its products so as to eliminate a fair use defense would 'substantially impede' the public's right to fair use and allow the copyright holder, through state law, to protect material that the Congress has determined must be free to all under the Copyright Act.[419]

Meanwhile, a state can permit parties to contract away a fair use defense if the contract is freely negotiated, which represents the "extra element."[420] Yet, a shrink-wrap license attempting to bind the purchaser to the software vendor's terms offers the only choice to avoid such in not making the purchase in the first place.[421] In other words, this self-proclaimed license provides a copyright holder with the superior power to override public law with private law in a contract of adhesion.[422] Thus, the shrink-wrap term eliminating the fair use defense should be unenforceable not because it is unconscionable but because it is not

---

415. *See Bowers*, 320 F.3d at 1325.

416. In a dissenting opinion, Judge Dyk disagreed with the majority approach that the contract claim is not preempted by federal law. *See id.* at 1335.

417. *See* Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992). *See generally* Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000); Step-Saver Data Sys., Inc. v. Wayse Techs., 439 F.2d 91 (3d Cir. 1991) (illustrating a split among the circuits).

418. *Bowers*, 320 F.3d at 1335 (quoting Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 157, 167 (1989)).

419. *Id.* at 1336 (Dyk, J., dissenting).

420. *Id.*

421. *See id.* at 1337.

422. *See* L. RAY PATTERSON & STANLEY W. LINDBERG, THE NATURE OF COPYRIGHT: A LAW OF USERS' RIGHTS 220 (1991) ("Such licenses are almost surely against public policy as unilateral attempts to override public law with private law in an adhesion contract.").

freely negotiated. Alternatively, the Federal Circuit should have merely reached the judgment that Baystate's reverse engineering had infringed on copyright, if the Court found substantial similarities, rather than becoming involved in conflictive analysis with public policy.

In UCITA, the concept requiring an "opportunity to review" establishes a requirement that resolves many procedural issues preventing unfair surprise.[423] Section 112(a) provides standards for manifestation of assent to which having an opportunity to review a record is a precondition:

> A person manifests assent to a record or term if the person, acting with knowledge of, or after having an opportunity to review the record or term or a copy of it: (1) authenticates the record or term to adopt or accept it; or (2) intentionally engages in conduct or makes statements with reason to know that the other party or its electronic agent may infer from the conduct or statement that the person assents to the record or term.[424]

V.   ANTITRUST STANDARDS FOR RESTRICTIONS ON REVERSE ENGINEERING

A.   *Antireverse-Engineering Contracts Under § 1 of the Sherman Act*

A provision prohibiting reverse engineering may be subject to antitrust scrutiny under § 1 of the Sherman Act. Although no antitrust claim regarding an antireverse-engineering provision has been brought until now, a number of potential antitrust issues will arise. Section 1 forbids any "contract, combination or conspiracy, in restraint of trade."[425] The offense of § 1 requires proof of three elements: (1) concerted action that (2) restrains trade and (3) is competitively unreasonable.[426] The first test for the § 1 cases is whether the alleged conduct is concerted action or merely the unilateral conduct of separate actors. Unlike the unilateral action of § 2, § 1 requires duality of action.[427] Second, the restraint of trade covers a variety of interpretations of business conduct. Antireverse-engineering clauses may constitute restraints of trade within the meaning of the statute, since they limit access to information about computer

---

    423.   *See* UCITA § 112 cmt. 8 (2002).
    424.   *Id.* § 112(a)(1)-(2).
    425.   15 U.S.C. § 1 (1982); *see also* RESTATEMENT (SECOND) OF CONTRACTS § 188, cmt. d (1981).
    426.   *See, e.g.*, Nat'l Soc'y of Prof'l Eng'rs v. United States, 435 U.S. 679, 690 (1978); Standard Oil Co. v. United States, 221 U.S. 1, 65 (1911).
    427.   *See* Fisher v. City of Berkeley, 475 U.S. 260, 267 (1986) (holding that a series of unilateral actions mandated by a municipal ordinance did not become "concerted" action within the meaning of § 1).

programs necessary to develop comparable products or prohibit buyers or licensees from copying object codes of the relevant software.

Contractual limitation, however, in itself will not be sufficient to prove antitrust violation; the nature of restraint should be characterized as unreasonable. This Part mainly focuses on the third test under § 1. It is a very difficult task because no coherently applicable and judicially administrable test can measure the "reasonableness" of a specific restraint. Courts over time, however, have identified two clearly defined standards for judging restraints: "per se illegality" and the "rule of reason." Per se illegal practices are viewed as so inherently anticompetitive as to be held illegal regardless of the reasons for or the effects of any particular use.[428] Such practices include price fixing, horizontal market allocation and horizontal boycotts. Practices not considered per se illegal are generally analyzed by the far more flexible balancing analysis known as the rule of reason. This approach weighs a broad inquiry into the nature, the defendant's intent and purpose, justifications for the restrictions, competitors' competitive position, and effect of any challenged arrangement.[429]

Academics and courts have also established several distinctive sets of standards which employ the "per se rule" or "rule of reason" labels in diverse ways: the traditional six-cell approach,[430] the Old White-Brandeis Rule of Reason,[431] Justice Peckham's mode,[432] and Judge Taft's Rule of

---

428. *Socony-Vacuum* was the first case to use the term "per se" as antitrust violations. *See* United States v. Socony-Vacuum Oil Co., 310 U.S. 150 (1940); WILLIAM C. HOLMES, ANTITRUST LAW § 2:8 (2002).

429. *See* LAWRENCE A. SULLIVAN & WARREN S. GRIMES, THE LAW OF ANTITRUST: AN INTEGRATED HANDBOOK § 5.1a (2000).

430. Courts classify agreements according to two characteristics—vertical agreement and horizontal agreement. Then courts also classify these structures as price restraints, nonprice restraints, and boycotts. All horizontal restraints are per se illegal, as are vertical price restraints. Vertical nonprice restraints and boycotts are subject to the rule of reason. However, the problem with this analysis is its failure to define reasonableness. *See* Peter C. Carstensen & Richard F. Dahlson, *Vertical Restraints in Beer Distribution: A Study of the Business Justification for and Legal Analysis of Restricting Competition*, 1986 WIS. L. REV. 1, 63-64 (1986).

431. In this approach, every restraint should receive an open-ended review that evaluates any and all justifications presented on its behalf as well as the full range of social costs that it might impose. By balancing these costs and benefits, the court determines the reasonableness of the restraint. *See* United States v. Trans-Mo. Freight Ass'n, 166 U.S. 290, 340-42 (1897); United States v. Joint Traffic Ass'n, 171 U.S. 505, 560-61 (1898).

432. Peckham suggested a strict dichotomy between direct (naked) and indirect (ancillary) restraints. Peckham held in the *Trans-Missouri* case that every contract "in restraint of trade," no matter how reasonable, violated the Act. Since he had no rule of reason in his approach, some ancillarity to the restraint apparently took it outside the law altogether. *See* United States v. Trans-Mo. Freight Ass'n, 166 U.S. 290 (1897).

Reason.[433] This Article applies the more conceptual functional approach developed by Professor Carstensen as the best analytical tool to evaluate restraints on reverse engineering.[434] According to this model, a key distinction between ancillary and naked restraints is central to understanding the judicial application of the rule of reason and the per se rule to restraints of trade.[435] If a restraint is naked, it can only be judged by per se rules: it need not be per se illegal; it may be per se legal.[436] However, if a restraint is ancillary, only then can it be judged as to its "reasonableness" in fact in a particular context.[437]

An ancillary restraint is defined as "a restriction or limitation which is functionally integral to some other legitimate, productive transaction or venture between the parties."[438] The salient function of the restraint is to facilitate or implement some aspect of the transaction or joint venture. Crucial to ancillarity is the finding that the parties are participants in some other transaction or productive activity which is legitimate and lawful.[439]

On the other hand, "[a] naked restraint is one whose only economic function is to restrict, limit, or affect the economic freedom of action of one or more parties thereto."[440] A naked restraint often results in good consequences such as lower transaction costs, avoidance of cut-throat competition, and improved safety. However, they flow exclusively as a consequence of the elimination of competition and are not a product of

---

433. Taft agreed with Peckham's approach that naked restraints are absolutely illegal. However, Taft argued that a restraint which was ancillary might also be unreasonable. *See* United States v. Addyston Pipe & Steel Co., 85 F. 271 (6th Cir. 1989).

434. *See* Carstensen, *supra* note 273, at 1.

435. The analysis of ancillary and naked restraints derives from the work of Robert Bork and William Howard Taft. *See* Robert Bork, *The Rule of Reason and the Per Se concept: Price Fixing and Market Division*, 74 YALE L.J. 775 (1965) (arguing that a restraint functions to facilitate some joint productive enterprise or transaction between parties, or to create, allocate, or exploit market power).

436. Some restraints are not illegal nor subject to § 1 of the Sherman Act, despite the manifest conclusion that the conduct at issue would be per se illegal. *See* Nat'l Collegiate Athletic Ass'n v. Bd. of Regents of Univ. of Okla., 468 U.S. 85 (1984); Silver v. N.Y. Stock Exch., 373 U.S. 341 (1963); Fed. Mar. Comm'n v. Seatrain Lines, Inc., 411 U.S. 726, 727-29 (1973); Carnation Co. v. Pac. Westbound Conference, 383 U.S. 213, 217-18 (1966) (finding that Congress vested the Shipping Board with authority to approve or disapprove of arrangements. Approval provided an exemption of liability from antitrust laws).

437. *See* Carstensen, *supra* note 273, at 1.

438. *Id.* at 2.

439. *Id.*

440. *Id.* at 4. In these categories, assigning territories or classes of customers, or agreeing to the characteristics of products that each will sell are all examples of naked restraints. A collective agreement to suppress relevant product information is also a naked restraint. *See id.*

any other transaction or joint productive activity involving the parties to the agreement.[441]

As Professor Carstensen notes:

> The analysis of ancillary restraint proceeds along the three steps: first, one defines the transaction or venture to which the restraint is ancillary and asks if that is a legitimate activity such that a restraint may be justified to achieve. Second, one asks whether the restraint functions to make that primary activity possible or more effective or successful. Third, one tests it for "reasonableness" by asking whether some less restrictive alternative would have made possible the same primary activity or transaction.[442]

In applying this antitrust analysis to various reverse-engineering situations, the relationship between rights conferred by IP that authorizes restraints on competition in the interest of exploiting the economic power of the rights and the need for restraint arising from a legitimate joint venture is intricate. An IP owner, patent or copyright, has certain rights to exclude competition that inhere in the grant of the right. The use of such rights is per se lawful if within the scope of the right. Hence, the definition of rights is very important. In other words, the fair use doctrine will play a crucial role in classifying naked restraints: per se legality or per se illegality, if there is no underlying transaction or joint venture among parties to the agreement. It is per se legal where a collusive agreement seeks to prohibit reverse engineering for the purpose of preventing development of products that contain copyrighted parts of the program. The antitrust law is not applicable to such restraints that are justified under copyright law forbidding copying protected materials.

However, it is naked, per se illegal, if a restraint that the software vendors impose functions to limit the reverse-engineering rights of the buyer or licensee, which is allowed under the fair use doctrine. Such a restraint, in fact, has an unreasonable effect: a prohibition on reverse engineering prevents discovery of unprotected ideas in a computer program; it functions to extend copyright protection beyond the exclusive rights granted to copyright holders under copyright law; and it may stifle technological development because competitors are restricted from developing compatible computer products.[443] Therefore, a restraint is naked, per se unlawful, to the extent that reverse engineering is permitted for certain uses under the fair use doctrine. According to Professor Carstensen, "efficiency arguments can be made on behalf of both naked and ancillary restraints because the difference between the two concepts

---

441. *See id.*
442. *Id.* at 8.
443. *See* Rogers, *supra* note 97, at 92-93.

is not their arguable efficiency, but the way or manner in which they achieve that efficiency."[444]

Meanwhile, an IP right holder may well joint venture with another enterprise to produce some good or service where the risks of opportunism or the inherent needs of the venture or transaction require restraint on competition. This restraint is ancillary to the legitimate joint venture. In a joint venture situation, any restraint which is within the scope of the lawful IP right should be in itself lawful, even if it is not the least anticompetitive restraint reasonably necessary for the primary goal. It is a harder question whether a restraint that is ancillary to the venture but which is in conflict with the limits on IP rights such as prohibiting reverse engineering of copyrighted material, is lawful in the context in which it is in fact ancillary and reasonable in terms of the joint venture. It is lawful on the same basis that the law forbids any restraint except when it is excused. In the case of IP, there are two excuses: property right and ancillarity. Whichever is more inclusive dominates in any context. Since antitrust laws aim to protect competitive process in markets, the patent or copyright laws provide appropriate protection against misconduct absent the competitive harm.[445]

Hypothetically, let us say that two companies are willing to make a joint venture to create new computer software that needs to develop functionality supporting the distinctive copyrighted programs owned by the two companies to work together. To achieve interoperability of the programs, the parties should disclose all information about the software and cooperate. They impose certain restrictions on the joint venture agreement because the companies are aware of the opportunistic risk to defect from the cooperation by one side. The agreement requires them not to disclose or license their own program to a third party until the venture ends. It also prohibits them from reverse engineering in the case of fraud by one party. This restraint might be ancillary because the function of restriction primarily makes the joint venture possible and more effective or successful. The relationship between the joint venture and copyrighted works is a significant factor as a matter of antitrust law, but has little relevance to copyright protection. The limitation on the freedom of the parties is not what copyright law suppresses. In addition, the restraint might be reasonable if there is not a less restrictive alternative. The cost of the restraint is the least that can be paid to obtain the benefits of the primary objective.[446] Thus, it is a beneficial restraint.

---

    444.   Carstensen, *supra* note 273, at 2.
    445.   *See* Walker Process Equip., Inc. v. Food Mach. & Chem. Corp., 382 U.S. 172 (1965).
    446.   *See* Carstensen, *supra* note 273, at 8.

This means that restraints, in excess of those legally authorized by IP in support of naked restraints on competition, such as shrink-wrap licenses imposed in connection with the bare sale of a product, are illegal under antitrust law. The relationship between a software vendor and buyers is merely final, not a continuous transaction. The antireverse-engineering provision unreasonably limits the buyers' rights under the fair use doctrine. Only if IP law is modified to alter or expand the scope of restraint authorized to an IP holder, can such conduct be lawful.[447]

The anticompetitive consequences of a restraint often rely on market power. In general, the more powerful the parties are in the market, the more likely their restraint will be condemned.[448] Therefore, if firms with monopoly power resulting from a copyrighted software product prohibit reverse engineering, that restraint is presumed anticompetitive. The fact that the software vendor possesses a copyright does not in itself indicate that it enjoys market power.[449] In contrast, the restraint involving firms with little market power is prima facie lawful. However, this market power approach does not seem to be very useful in identifying restraints because such a presumption of unlawfulness may be rebutted, and showing market power is a difficult task.[450] This can usually be done by establishing the business necessity of a restraint.[451] In some cases, however, showing market power is crucial and can lead to antitrust abuse.[452]

For example, firms holding a standard computer program come into agreement on modification of the standard routinely and trivially for the purpose of prohibiting competitors from developing products that are compatible with the program. They also agree to forbid reverse engineering. Even though some competitors independently achieve compatibility with the standard without reverse engineering, such slight modifications   suddenly   make   competitors'   products   no   longer

---

447. This is a strict standard and conflicts with the intuition of some courts that contract in connection with the sale can include such restraints.

448. *See* ABA ANTITRUST SECTION, MONOGRAPH NO. 2, VERTICAL RESTRICTIONS LIMITING INTRABRAND COMPETITION 63-64 (1977).

449. *See* Abbott Labs. v. Brennan, 952 F.2d 1346, 1354-55 (Fed. Cir. 1991) (finding no presumption of market power from intellectual property right); U.S. DEP'T OF JUSTICE & FED. TRADE COMM'N, ANTITRUST GUIDELINES FOR THE LICENSING OF INTELLECTUAL PROPERTY [hereinafter IP GUIDELINES], *at* http://www.usdoj.gov/atr/public/guidelines/ipguide.htm#t322 (last visited Dec. 6, 2002).

450. *See* Gough v. Rossmoor Corp., 585 F.2d 381, 389 (9th Cir. 1978).

451. *See id.*; Carstensen & Dahlson, *supra* note 430, at 66.

452. *See* SULLIVAN & GRIMES, *supra* note 429, § 5.1b ("[A] *per se* rule that requires no proof of market power is still warranted when a quick look discloses an abuse routinely associated with an exercise of market power and devoid of efficiency potential.").

compatible.[453] The primary function of restraint is impeding legitimate economic activities by others rather than improving the underlying technology. It ultimately proves worse for competitors to maintain compatibility even without reverse engineering. This is a plausible example of naked restraint associated with an exercise of market power.

B.   *Conceivable § 2 Issues on Reverse Engineering*

    An antitrust approach to issues of software reverse engineering will have different features from a traditional one under § 2 of the Sherman Act in terms of monopoly power, market definition, monopolistic conduct, business justifications, and so on because of the complexity of computer software associated with IP laws. This Part examines only scattered but highly potential situations arising under § 2 since no antitrust litigation has been involved in such issues. Section 2 of the Sherman Act prohibits efforts to "monopolize, or attempt to monopolize" any part of the trade.[454] To establish monopolization, a plaintiff must show market power in a relevant market and exclusionary conduct such as the willful acquisition or maintenance of that power.[455]

1.   Market Power Under Software Market Definition

    Estimating market power generally entails definition of a relevant market that is critical to determining an antitrust violation. Market definition aids in measuring monopoly power over a relevant market where a monopolist controls prices and output, or excludes competition.[456] Alternatively, the direct proof approach may be employed to determine monopoly power even in the absence of elaborate market analysis if evidence indicates that a firm has in fact sustained adverse effects on competition in areas where it predominates, or where it has raised prices substantially above the competitive level.[457] Thus, once a plaintiff shows such a direct evidence of power, the defendant has the

---

    453.   *See* Morrison, *supra* note 226, at 304.
    454.   15 U.S.C. § 2 (1988).
    455.   *See* United States v. Grinnell Corp., 384 U.S. 563, 570-71 (1966).
    456.   *See* United States v. E.I. du Pont de Nemours & Co., 351 U.S. 377, 391 (1956); Ball Mem'l Hosp. v. Mut. Hosp. Ins., Inc., 784 F.2d 1325, 1335 (7th Cir. 1986).
    457.   *See* United States v. Microsoft Corp., 253 F.3d 34, 56-59 (D.C. Cir. 2001); Rebel Oil Co. v. Atl. Richfield Co., 51 F.3d 1421, 1434 (9th Cir. 1995); FTC v. Indiana Fed'n of Dentists, 476 U.S. 447, 460-61 (1986) (using direct proof to show market power in a § 1 claim); Nat'l Collegiate Athletic Ass'n v. Bd. of Regents of Univ. of Okla., 468 U.S. 85, 109-10 (1984) (holding that "[a]s a matter of law, the absence of proof of market power does not justify a naked restriction on price or output," and that such a restriction "requires some competitive justification even in the absence of a detailed market analysis").

substantial burden of explaining why the market power is reasonable.[458] However, absent such direct evidence, courts typically examine market structure in search of circumstantial evidence of monopoly power.[459] Under this structural approach, monopoly power may be inferred from a firm's possession of a dominant share of a relevant market that is protected by entry barriers.[460]

Courts traditionally define a relevant market in terms of substitutability of products and geographic space where competition, in fact, exists. [461]   In the Cellophane case, the Supreme Court used the economic concept of cross-elasticity of demand as a standard to define a product market.[462]   Products are considered to be in the same market if they are reasonably interchangeable in use and if their prices are reasonably comparable.[463]   The Court held that cellophane and other flexible packaging material were in the same market because there was a high cross-elasticity of demand between these products.[464]   The federal antitrust agencies, in the U.S. Department of Justice's Merger Guidelines,[465] have articulated this principle by attempting to identify the group of goods or services which are reasonable substitutes for each other. They ask whether a "small but significant and nontransitory" price increase by a hypothetical monopolist would cause customers to switch to other goods or services. When a price increase of five percent per year lasts for the foreseeable future, the products to which the customers would switch should be included in the market and the inquiry made again.[466]

A similar approach is applied to definition of the geographic market. The Supreme Court has stated that the market should be defined by identifying the area in which the seller operates and within which customers can reasonably look for alternative sources of supply if the price of the good or service increases in a modest but lasting way.[467]

---

458.   *See* Image Technical Services, Inc. v. Eastman Kodak Co., 504 U.S. 451, 469 (1992).

459.   *See Microsoft Corp.*, 253 F.3d at 51 (rejecting Microsoft's assertion that, because the software industry is uniquely dynamic, direct proof of market power rather than the standard circumstantial evidence was more appropriate); 2A AREEDA & HOVENKAMP, ANTITRUST LAW ¶ 531a, at 156 (1995).

460.   *See Rebel Oil*, 51 F.3d at 1434.

461.   *See* United States v. E.I. du Pont de Nemours & Co., 351 U.S. 377, 393-95 (1956); Brown Shoe Co. v. United States, 370 U.S. 294, 328 (1962).

462.   *See E.I. du Pont de Nemours & Co.*, 351 U.S. at 395.

463.   *See id.*

464.   *See id.* at 394-95, 400.

465.   U.S. DEP'T OF JUSTICE & FED. TRADE COMM'N, MERGER GUIDELINES, *at* http://www.ftc.gov/bc/docs/horizmer.htm (last visited Dec. 6, 2002).

466.   *See id.* § 1.11.

467.   *See* Tampa Elec. Co. v. Nashville Coal Co., 365 U.S. 320, 327 (1961).

Hence, market definition has become a definitive issue on trial because a broadly defined product or geographic market will generally produce lower market shares, while a narrowly defined market will relatively underestimate the relevant elasticities.[468]

The most difficult questions in a market definition inquiry revolve around market dynamics caused by technological and other changes such as those in the field of computer software. For example, if a certain software product that is currently in a different market from a dominant operating system will have enough capacities to take over operating system functions, then how much change or improvement is enough for the product to be considered in the relevant market? To what extent is a line drawn to define relevant markets in order to protect software innovations directed to develop a good that is far away from a present market? Indeed, there has been much fear in such an industry that the pace or direction of R&D and competition can be adversely affected by the monopolistic conduct. The traditional product market has not sufficiently dealt with these concerns. Therefore, the FTC and the DOJ have adopted the concepts of "technology" and "innovation" markets through the IP Guidelines,[469] which are valuable instruments in evaluating competitive effects on innovation associated with copyrighted computer software and its licensing arrangements. Defining technology and innovation markets recognizes the significant value of IP and the complexity of licensing arrangements, while the traditional market focuses on market competition for the finished goods or services. A technology market is defined as a market comprised of the actual intellectual property that is being licensed, transferred, or acquired and any technologies and goods that are considered close substitutes.[470] This technology market is utilized to determine whether a licensor of intellectual property has market power which would enable anticompetitive conduct.[471]

On the other hand, since some competitive effects of licensing arrangements cannot be adequately addressed through the analysis of goods or technology markets, the concept of an innovation market gives

---

468. Based on the defendant's market definition, du Pont had only 17.9% of the broader market including other flexible wrapping materials even though it controlled 75% of the cellophane market. *Id.* at 379, 399; Todd W. Miller, Market Analysis in Merger, Monopoly and Entry in Telecommunication, at 2 (Oct 2-4, 2000) (paper presented at the Antitrust in Telecommunications Markets Workshop at the University of Wisconsin-Madison School of Business); RICHARD A. POSNER, ECONOMIC ANALYSIS OF LAW 326 (5th ed. 1998).

469. *See* IP GUIDELINES, *supra* note 449, §§ 3.2.1-3.2.3.

470. *See id.* § 3.2.2.

471. *Id.*

increased legitimacy to R&D analysis. The *U.S. Department of Justice and Federal Trade Commission Antitrust Guidelines for the Licensing of Intellectual Property* defines an innovation market as the "research and development directed to particular new or improved goods or processes, and the close substitutes for that research and development."[472] The innovation market concept is important when licensing arrangements have competitive effects on an innovation that is directed at a goods market, which presently does not exist.[473] The arrangements may also affect the development of new or improved goods or processes in geographic markets where there is no actual or likely potential competition in the relevant goods.[474] Commentators criticize the innovation market theory because not only do the IP Guidelines provide little insight into how the analysis is to be done, but enforcement resources are scarce as well.[475] Indeed, no theoretical and empirical studies support the proposition that increased concentration yields decreased innovation.[476]

Nevertheless, the concepts of technology market and innovation market are valuable to evaluate the competitive effects of antireverse engineering on future generation markets, especially separate or complementary markets. For example, suppose a defendant who dominates in the primary market intends to use its power to prevent creation of competition in a new separate market because it seeks to enter and dominate the new market. The defendant has practically no market

---

472. *Id.* § 3.2.3.

473. *See, e.g.*, Ciba-Geigy Ltd., Docket No. C-3725 (March 24, 1997); Sensormatic Electronics Corp., FTC Inv. No. 941-0126, 60 Fed. Reg. 5428 (accepted for comment Dec. 28, 1994); Wright Medical Technology, Inc., FTC Inv. No. 951-0015, 60 Fed. Reg. 460 (accepted for comment Dec. 8, 1994) (involving the acquisition of a firm engaged in the research and development of an improved orthopaedic implant by the firm producing the current generation of implants); American Home Products, FTC Inv. No. 941-0116, 59 Fed. Reg. 60,807 (accepted for comment Nov. 28, 1994) (examining a complaint alleging "the research and development of a vaccine against Rotavirus infection in humans" as a market).

474. *See* IP GUIDELINES, *supra* note 449, § 3.2.3 (citing United States v. Gen. Motors Corp., Civ. No. 93-530 (D. Del., filed Nov. 16, 1993)).

475. *See* Mary L. Azcuenaga, Antitrust and Intellectual Property: Recent Highlights and Uncertainties, Remarks at the meeting of the American Law Institute-American Bar Association (Apr. 24, 1997), *at* http://www.ftc.gov/speeches/azcuenaga/aliaba97.htm (last visited Dec. 2, 2002); Michael N. Sohn, Competitive Effects and Entry Analysis in R&D and Future Generation Markets, Testimony Outline at the FTC Hearings on Global and Innovation Based Competition (Oct. 25, 1995), *at* http://www.ftc.gov/opp/global/sohn2038.htm (last visited Dec. 2, 2002); Azam H. Aziz, *Defining Technology and Innovation Markets: The DOJ's Antitrust Guidelines for the Licensing of Intellectual Property*, 24 HOFSTRA L. REV. 475 (1995).

476. *See* Azcuenaga, *supra* note 475, at 8. Because it is extremely difficult to predict future competitive potential in a relevant market, most analysis tends to be somewhat more static. *See* Miller, *supra* note 468, at 3.

share at all of the potential separate product market; it is only a new entrant.[477]   In the absence of market power, most forms of business behavior other than price fixing are legal.[478]   However, from the viewpoints of technology and innovation market theory, the defendant might have market power relative to developing a new product, if it substantially invests in R&D and has relevant IP licensing arrangements and restrictive contracts. The firm may effectively restrict others from making productive investment or R&D in the separate markets, which might have brought the consumer a better product, if reverse engineering the monopolist's primary software is essential to the innovation process. Hence, the concepts of innovation market and technology market still provide an adequate platform for appraising exclusionary practices in an anticompetitive manner.

   Just as significantly, "network effects" appear in most innovation and technology markets. "Actual network effects" based on direct communications systems (e.g., the wireline telephone industry, Instant Messaging, the Internet industry, etc.) have long been regarded as the classic network effect. One customer's telephone is more valuable as more people are added to the system, thereby allowing the customer to connect with more people. On the other hand, "indirect or virtual network effects" illustrate that the more consumers own identical or interoperable goods, the more benefit each consumer derives from service or accessibility. The computer software industry exhibits obvious virtual network effects. Unless the new software is original or highly specialized, most software developers attempt to write a program compatible with an industry standard because of its widespread acceptance in the market.[479] On this point, the value of competition in the software industry grows when more compatible goods are able to enter into the market.[480] Hence, the broadened scope of the reverse engineering right serves to encourage dynamic competition and intellectual activities, at least in the network industry.

---

   477. *See* Lande & Sobin, *supra* note 18, at 276. In cases where actual market share in the separate market is low, a leveraging theory may fail only under the end-product market approach. *Id.*

   478. *See* THOMAS M. JORDE & DAVID J. TEECE, IN ANTITRUST, INNOVATION, AND COMPETITIVENESS 7 (1992).

   479. *See* Joseph Hinshaw, *The Role of Standardization and Interoperability in Copyright Protection of Computer Software*, 4 COMM. L. & POL'Y 299, 299 (1999).

   480. *See* LEMLEY & MCGOWAN, *supra* note 17, at 484.

2.   Unlawful Monopolistic Conduct

Possessing monopoly power does not itself violate § 2 of the Sherman Act: a monopolist must engage in exclusionary conduct to maintain its monopoly, which is distinguished from growth or development as a consequence of a superior product or business acumen.[481] A central tenet of any software antitrust case, such as that for software reverse engineering, is that a software vendor fights to crush other software developers because the vendor fears that the competitors' end works can one day be used as an alternative to the software. A copyright owner may attempt to put all kinds of restrictions on licensing contracts or create technological impediments to make sure that others are not able to develop products that are interoperable or comparable with the work.[482]

Technological protection or impediments along with other anticompetitive behaviors aimed at preventing reverse engineering may also raise § 2 issues. A software vendor can effectively prohibit buyers from reverse engineering by including technological measures in its software. Suppose there is a company that dominates a substantial market share with its operating system that has become industry standard software. Even though it has not been involved in the application market, it plans to develop a specific application for the Internet Web browser working on its platform system. To reinforce the marketability of the new application, the company refuses to reveal to existing or potential competitors in the Web browser market any information about the operating system necessary to develop products compatible with the system. It also embodies specialized technological protection, lock-and-key measures in its operating program, which effectively impede reverse engineering. At the very same time, the monopolist fully discloses the information necessary to work around these obstacles to its own software designers who are writing the Web browser program.

The technological means against reverse engineering would make it more difficult for a rival's software to be developed and distributed.[483] Such means, in fact, have so-called technological tying effects: buyers of the operating system who want a compatible application program are essentially forced to purchase a Web browser developed by the

---

481.   *See* United States v. Grinnell Corp., 384 U.S. 563, 570-71 (1966).
482.   However, a copyright owner may want to encourage interoperability in order to foster the creation of a virtual network effect.
483.   *See* Bender & Johnston, *supra* note 21, at 745. Technological impediments should be subject to § 2 concerns since they may achieve the same effect as a contractual prohibition on reverse engineering—tying is one example among many possible situations. *Id.*

monopolist.[484] As explained above, a tying occurs when a party sells a product or service on the condition that the buyer also purchases another product or that the buyer does not deal in the goods of competitors.[485] In *Jefferson Parish Hospital District No. 2 v. Hyde*, the Supreme Court established a modified per se approach to examine tying practices.[486] The rule requires the plaintiff to establish three distinctive elements: (1) a tying arrangement exists between two separate products; (2) the "seller has some special ability—usually called 'market power'—to force a purchaser to do something that he would not do in a competitive market"; and (3) the tying arrangement forecloses a substantial volume of commerce.[487]

     Distinguishing a tie from a procompetitive bundling is a very difficult question when it comes to integration of two separate computer software programs using a common interface. In general, if integration is reasonably disciplined by competition or otherwise within the confines of informed consumer demand, there is no tie.[488] For example, bundling the developed Web browser with the industry standard operating system can improve the efficiency or significantly lower the marketing costs of both products. However, if the bundling of two products is employed as a tool to put competitors in the relevant market at a disadvantage, it should be considered a tie. The disputes between Microsoft and the U.S. government on efficiency claims in integrated browsers are a famous case in point.[489]

     The district court in the *Microsoft* case applied the modified per se rule to find that Microsoft's tying of its Web browser (the tied product) with the Windows operating system (the tying product) violated § 1 of the Sherman Act. However, the court of appeals rejected the district court's per se analysis, holding that the rule of reason should govern the legality of tying involving high-technology or software product-integration.[490] Because the issue of software bundling is new, there is a high risk that the per se approach may produce inaccurate results such as ignorance of the possible procompetitive effects. The court was especially very skeptical about the separate-products test that would focus on consumer demand in the analysis of the efficiencies of a

---

     484.   *See* Lande & Sobin, *supra* note 18, at 279.
     485.   *See* SULLIVAN & GRIMES, *supra* note 429, at 387.
     486.   466 U.S. 2 (1984).
     487.   *Id.* at 13-14 (citations omitted).
     488.   *See* Eastman Kodak Co. v. Image Technical Servs., 504 U.S. 451, 462 (1992); SULLIVAN & GRIMES, *supra* note 429, at 390-91.
     489.   *See* United States v. Microsoft, 253 F.3d 34 (D.C. Cir. 2001) (en banc).
     490.   *See id.* at 84-95.

bundled sale.[491] The court found that the rule of reason analysis rather than the inquiry based on per se rule "afford[ed] the first mover an opportunity to demonstrate that an efficiency gain from its 'tie' adequately offsets any distortion of consumer choice."[492] However, the court of appeals' approach is not likely to change outcomes, but is likely to impose a heavy burden on plaintiffs establishing an illegal tying. In fact, the separate-products test allows inquiry into supply-side as well as demand-side efficiency.[493] The Supreme Court in *Kodak II* held that it was efficient to provide service separately from parts because not only might there be sufficient consumer demand, but also the development of the entire high-technology service industry was evidence of the efficiency of a separate market for service.[494] Thus, a seller could argue that a bundled package produces more efficiencies than separate components, which benefits consumers through improved quality, convenience, or ease of operation.

Altering the interfaces of the operating system for the purpose of disadvantaging competitors in a separate market, as in the example previously discussed for § 1 issues, could be another possible technological impediment associated with software reverse engineering. In 1981, the European Commission brought suit against IBM for an abuse of a dominant position in Europe.[495] It characterized the following as abuses: bundling its software with hardware, frequent changing of interfaces and announcing new products to customers without timely disclosure of the new interfaces and certain other practices that IBM abandoned, which disadvantaged European makers of peripheral products. The dispute was eventually resolved by IBM's agreement to announce changes to its interfaces in advance so that peripheral manufacturers could adjust their products accordingly.[496] Thus, if a

---

491. *See id.* at 92 (finding that the "separate-products test is a poor proxy for net efficiency from newly integrated products").

492. *Id.* (citation omitted).

493. *See* Warren S. Grimes, *The Antitrust Tying Law Schism: A Critique of Microsoft III and a Response to Hylton and Salinger*, 70 ANTITRUST L.J. 199, 206-07 (2002) ("Even if a first-mover could not establish that its bundled sale was a single product, application of the modified per se rule could be defeated by establishing the quality control defense available to a new or substantially modified product."); HERBERT HOVENKAMP, FEDERAL ANTITRUST POLICY: THE LAW OF COMPETITION AND ITS PRACTICE § 10.5b, at 411 (2d ed. 1999) (stating that the defendant may establish that a bundled sale is a single product by showing that the seller has to tie the innovated items in order to ensure that they work effectively together).

494. Eastman Kodak Co. v. Image Technical Servs., Inc., 504 U.S. 451, 462 (1992).

495. Int'l Bus. Machs., v. Commission, 1981 E.C.R. 1857.

496. *See* COMM'N OF THE EUROPEAN CMTYS., FOURTEENTH REPORT ON COMPETITION POLICY § 6, points 95 (1985); [1982-1984 New Developments] COMMON MKT. REP. (CCH) ¶ 10,608 (1984); Samuelson & Scotchmer, *supra* note 242, at 1656 n.373.

software vendor's strategy—technological change along with prohibition on reverse engineering—does not derive exclusively from the merits of its product, but unreasonably excludes competitors from relevant markets, it should be subject to antitrust concerns.

      Technological impediments on reverse engineering might often be accompanied by contract restrictions which add up to anticompetitive behavior.   Suppose a small company whose main business is manufacturing personal computers plans to enter a Web browser market. Its software programmers decide to write a Web browser program for the most popular operating system because application software will only function with one particular operating system. Thus, it seeks to obtain the functions of the industry standard operating system possessed by a large company, which are necessary for the Web browser to operate on the system.   However, the terms of the dominant firm's license agreement restrict any reverse engineering of the operating system. It also threatens the PC maker under the arrangement whereby it will not offer its operating system unless the small company stops producing its Web browser and rather adopts a Web browser developed by the large company.

      The restrictions imposed by the large firm in licensing its operating system prevent the PC maker from developing and distributing its browser as an alternative to the large company's browser. A dominant firm cannot illegally use its muscle in one market to defend and extend the reach of its monopoly to another market. Despite the fact that no market share is yet possessed by the large firm in the separate market, the conduct may constitute exclusionary restraints if no obvious major efficiency exists. The large company, which has market power in the operating system market, employs contractual restrictions to eliminate potential competition in the separate Web browser market it plans to enter.[497] Meanwhile, assume the large firm has already had some market power in the separate market and seeks to preserve and enhance such power. This is a classic leveraging antitrust case.

      The illegal monopolistic conduct can be rebutted by proffering legitimate business justifications for the restriction on reverse engineering. A valid copyright holder typically argues that it is simply exercising its exclusive rights; the copyright itself justifies otherwise anticompetitive behavior. As discussed above, IP rights do not confer

---

      497. *See* United States v. Griffith, 334 U.S. 100, 107 (1948) ("[T]he use of monopoly power, however lawfully acquired, to foreclose competition, to gain a competitive advantage, or to destroy a competitor, is unlawful.").

permission to violate the antitrust laws.[498] Meanwhile, a software vendor might provide the peculiar economy of software markets as a justification for denial of access. Software markets are so dynamic that the product lifecycles are shorter than in other markets. Thus, protecting software from being copied and large up-front investment in development is extremely significant.[499] This query should turn to the scope of copyrights: specifically, it is important to determine whether the restriction is within the scope of the fair use doctrine. If the action of a monopolist has the effect of limiting buyers or licensees' fair use of copyrighted material, then the copyright vendor should not be exempted from antitrust condemnation.

The software vendor, however, may limit a licensee's reverse engineering to engage in substantial and deleterious alterations of a copyrighted work.[500] If the license restrictions merely prevent buyers or licensees from taking actions that would substantially reduce the value of a copyrighted work, the restrictions are not an exclusionary practice.[501] For example, suppose the original equipment manufacturers (OEMs) set some program upon their PC system so that it automatically replaces the copyrighted components of a program with alternative interfaces which they have designed. In such a situation, a copyright holder might impose limitations on reverse engineering of the current or advanced versions of its product unless the OEMs redeem their actions.

If a defendant's restrictions are redeemed by legitimate justifications, a plaintiff would bear the burden of rebutting the proffered justification.[502] As for the software reverse engineering issues, this strikes a balance under the alternative standard test. In a case where the defendant's software is essential for the plaintiff to develop a competitive product, the plaintiff can rebut the justification by showing that there is "a less restrictive alternative that would accomplish the same legitimate goal or protect against a valid risk."[503] If there is an alternative way that would not result in exclusionary treatment of the plaintiff, the restraints at issue should be condemned. In other words, the monopolistic practice should not necessarily be anticompetitive and destroy future choices that will benefit consumers.

---

498.   *See In re* Indep. Serv. Orgs. Antitrust Litig., 203 F.3d 1322, 1325 (Fed. Cir. 2000).

499.   *See* Lande & Sobin, *supra* note 18, at 268.

500.   *See* Gilliam v. ABC, 538 F.2d 14, 21 (2d Cir. 1976).

501.   *See* United States v. Microsoft, 253 F.3d 34, 63-64 (D.C. Cir. 2001) (en banc).

502.   *See id.* at 67. The plaintiff should demonstrate that the anticompetitive effect of the challenged action outweighs the justification. *Id.*

503.   Carstensen, *supra* note 273, at 9.

3.   Attempted Monopolization

The offense of attempted monopolization requires a plaintiff to prove: (1) specific intent to monopolize, (2) anticompetitive conduct, and (3) a dangerous probability of success.[504] Unlike monopolization, attempted monopolization is a specific intent offense: the defendant intended that its conduct, if successful, would give it monopoly power in a properly defined market. Even though the challenged conduct has not yet succeeded in achieving monopoly power, there must be a dangerous probability that it would do so if not stopped by the court.[505] Because unlawful activity may or may not be sufficiently dangerous to justify condemnation, an intent requirement was thought necessary to ensure that the defendant posed a sufficient danger to society to justify punishment.[506] The dangerous probability of success requirement is satisfied only when prohibition on reverse engineering has the potential to harm competition, and not merely because the conduct is intrinsically against the fair use doctrine. The anticompetitive conduct element is identical to that required under the offense of monopolization. Thus, the simple attempt to attain monopoly power cannot be prohibited; the defendant must have engaged in conduct that, if successful, could be characterized as the "willful" acquisition of monopoly power.[507]

Attempted monopolization fills in the gap between § 1 conduct and illegal monopolization under § 2: it can cover anticompetitive conduct by a single firm that does not yet have monopoly power.[508] Thus, the market share requirement is slightly lower than monopolization. Whereas monopolization cases usually requires market share in the range of seventy to one hundred percent,[509] attempted monopolization cases usually require only a fifty percent share, and sometimes less.[510]

VI. CONCLUSION

Greater protection of intellectual property has raised a significant access question in this era of high technology. Enforcement of a total prohibition on fair access to a computer program through reverse

---

504.   *See* Swift & Co. v. United States, 196 U.S. 375 (1905).
505.   *See id.* at 396.
506.   *See* STEPHEN F. ROSS, PRINCIPLES OF ANTITRUST LAW 100 (1993).
507.   *See* Photovest Corp. v. Fotomat Corp., 606 F.2d 704 (7th Cir. 1979), *cert. denied,* 445 U.S. 917 (1980).
508.   *See* ROSS, *supra* note 506, at 99.
509.   *See* REPORT OF THE ATTORNEY GENERAL'S NATIONAL COMMITTEE TO STUDY THE ANTITRUST LAWS 49 (1955).
510.   *See* Lande & Sobin, *supra* note 18, at 277.

engineering interferes with the federal copyright system because copyright protection is elevated to the level of patent protection. This is accomplished by providing protection over unprotected ideas and functional elements. Indeed, since reverse engineering plays a significant role in the exploitation of knowledge committed to the public domain through the grant of copyrights, prohibiting fair use of the copyrighted material by placing a black dot on a shrink-wrap license may stifle the drive to study and improve upon the existing knowledge base.[511]

In general, reverse engineering computer software for interoperability or comparability rather than for exploitation should be allowed if a final product contains as little as possible of these portions of the code from the original program, which are necessary to achieve interoperability or compatibility. In this regard, the term "inter-operability" of § 1201(f)(A) of the DMCA should be expansively interpreted in order to allow reverse engineering for a purpose other than interoperability if legitimate access is given or the central objective of reverse engineering is not to overcome the protective system protected by contract. By the same token, relevant sections of UCITA should be interpreted to comply with the reverse engineering policy of the federal copyright law: a shrink-wrap term which eliminates the fair use defense should be unenforceable.

Much of the discussion concerning competition criteria is meaningful in terms of conceiving the possible anticompetitive effects of forbidding reverse engineering rights. A copyright holder has certain rights to exclude competition. Any restraint which is within the scope of the lawful copyright should in itself be lawful, even if it is not the least anticompetitive restraint reasonably necessary for the primary goal. In contrast, when a restraint imposed by a software vendor functions to limit the reverse engineering rights, which is allowed under the fair use doctrine, the restraint may be subject to antitrust scrutiny. In this respect, a shrink-wrap license may be illegal under antitrust law when an antireverse engineering provision is in excess of its legally authorized scope of IP in support of naked restraints on competition. Limitations on reverse engineering rights along with other anticompetitive behavior also give rise to § 2 claims of the Sherman Act when a software vendor attempts to put all kinds of restrictions on licensing contracts or creates technological impediments to make sure that others are not able to create products which are interoperable or compatible with the work.

---

511. *See* Uhrich, *supra* note 26, at 149.