Fordham International Law Journal

Volume 14, *Issue* 4 1990 *Article* 6

The EEC Directive on the Legal Protection of Computer Programs and U.S. Copyright Law: Should Copyright Law Permit Reverse Engineering of Computer Programs?

Mindy J. Weichselbaum*

*

The EEC Directive on the Legal Protection of Computer Programs and U.S. Copyright Law: Should Copyright Law Permit Reverse Engineering of Computer Programs?

Mindy J. Weichselbaum

Abstract

This Note compares the reverse engineering exception in the Directive with U.S. copyright law. Part I sets forth the background of copyright law for computer software, with emphasis on the act of reverse engineering. Part II charts the evolution of the reverse engineering exception in the Directive, and the debate that this exception has spurred. Part III compares the Directive's reverse engineering exception with the U.S. copyright law that is applicable to reverse engineering. Part IV argues that U.S. law prohibits reverse engineering, and that the Community could better serve the European software industry by similarly granting broad copyright protection without a reverse engineering exception. This Note concludes that the inclusion in the Directive of an exception permitting unauthorized reverse engineering is misguided. Instead, Community legislation should confer broad copyright protection to the rightholder

THE EEC DIRECTIVE ON THE LEGAL PROTECTION OF COMPUTER PROGRAMS AND U.S. COPYRIGHT LAW: SHOULD COPYRIGHT LAW PERMIT REVERSE ENGINEERING OF COMPUTER PROGRAMS?

INTRODUCTION

Computer software, like other original works of authorship, requires legal protection from piracy.[1] The United States and the European Economic Community (the "EEC" or the "Community") both recommend that copyright law protect computer software.[2] The EEC Directive on the Legal Protection of Computer Programs (the "Directive")[3] will afford copy-

[1] *See* NAT'L COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 10 (1978) [hereinafter CONTU REPORT]; R. NIMMER, THE LAW OF COMPUTER TECHNOLOGY 9 (1985); *infra* notes 18-20 and accompanying text (discussing need to protect computer programs legally).

[2] *See* Council Directive on the Legal Protection of Computer Programs, No. 91/250, O.J. L 122/42 (1991) [hereinafter Directive]; *see also* 17 U.S.C. §§ 101-17 (1988); CONTU REPORT, *supra* note 1, at 11. The 1976 U.S. copyright statute did not contain any special limitations on computer programs. *See* 17 U.S.C. §§ 101-17 (1988). In 1978, the Commission on New Technological Uses of Copyrighted Works ("CONTU") issued recommendations to grant computer programs copyright protection. *See* CONTU REPORT, *supra* note 1, at 10-11. These recommendations led to the enactment of the U.S. Computer Software Act of 1980 which amended the 1976 act. 17 U.S.C. §§ 101 & 117 (1988). The 1980 act added the definition of computer program to the 1976 statute and added a section on the limitations of copyright with regard to computer programs. *See id.*

Like the United States, the European Economic Community (the "EEC" or the "Community") has recognized copyright as the protection for computer programs. *See* Directive, *supra*, art. 1, at 43. Article 1 of the Directive provides that

> 1. [i]n accordance with the provisions of this Directive, Member States shall protect computer programs, by copyright, as literary works within the meaning of the Berne Convention for the Protection of Literary and Artistic Works. For the purpose of this Directive, the term "computer programs" shall include their preparatory design material.
>
> 2. Protection in accordance with this Directive shall apply to the expression in any form of a computer program. Ideas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright under this Directive.
>
> 3. A computer program shall be protected if it is original in the sense that it is the author's own intellectual creation. No other criteria shall be applied to determine its eligibility for protection.

Id.

[3] *See* Directive, *supra* note 2. The Directive was proposed by the Commission of the European Communities (the "Commission") in 1989. *See* Proposal for a Council Directive on the Legal Protection of Computer Programs, O.J. C 91/4 (1989) [hereinafter Commission's Proposed Directive]. In July 1990, the Parliament of the European Communities (the "Parliament") recommended amendments to the Proposed

right protection to computer software authors.[4] Similarly, the U.S. Copyright Act of 1976 (the "U.S. Copyright Act" or the "Copyright Act")[5] as amended by the U.S. Computer Software Act of 1980[6] grants copyright protection with limited exceptions to software authors.[7]

A current issue in the application of copyright protection

Directive. EUROPEAN PARLIAMENT, PROPOSED AMENDMENTS (July 4, 1990) [hereinafter PARLIAMENT AMENDMENTS]. The Commission incorporated most of the terms in the Parliament's amendments in an amended draft of the Proposed Directive. Amended Proposal for a Council Directive on the Legal Protection of Computer Programs, O.J. C 320/12 (Oct. 1990) [hereinafter Amended Proposed Directive]. The Amended Proposed Directive was submitted to the Council of Ministers of the European Communities (the "Council"), which issued its Common Position Paper in December 1990. COUNCIL OF MINISTERS, COMMON POSITION PAPER, art. 1, at 7 (Dec. 14, 1990) (copy on file at the *Fordham International Law Journal* office) [hereinafter COMMON POSITION PAPER]. On May 14, 1991, the Council adopted the text of the Common Position Paper as the Directive. Directive, *supra* note 2, at 46.

[4] *See* Directive, *supra* note 2, art. 1, at 44.

[5] Pub. L. No. 94-553, 90 Stat. 2541 (1976), codified at 17 U.S.C. §§ 101-17 (1988).

[6] Pub. L. No. 96-517, § 10, 94 Stat. 3028 (1976), codified at 17 U.S.C. §§ 101-17 (1988).

[7] *See* 17 U.S.C. § 106 (1988). Section 106 grants exclusive rights to the copyright holder. *Id.* Section 106 provides that

> the owner of copyright under this title has the exclusive rights to do and to authorize any of the following:
>
> (1) to reproduce the copyrighted work in copies or phonorecords; (2) to prepare derivative works based upon the copyrighted work; (3) to distribute copies or phonorecords of the copyrighted work to the public 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; and (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.

Id. Section 117 places limitations on the rights in section 106 for computer programs. *Id.* § 117. Section 117 provides that

> [n]otwithstanding the provisions of section 106, it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided: (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or (2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.
>
> Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other

to computer software concerns whether the copyrighting of the program would prevent reverse engineering.[8] Reverse engineering can help create interoperable programs, which are beneficial to small programmers who want to link their programs to those of the larger manufacturers.[9] Reverse engineering, however, can also facilitate the act of copying programs, and open the door to piracy.[10]

The Directive explicitly permits reverse engineering as an exception to the exclusive rights granted by copyright protection.[11] Conversely, the status of reverse engineering under

> transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

Id.

[8] Bernier, *Software Producers Do Battle Over Proposed EC Directive: A Proposed European Community Directive Providing Legal Protection To Computer Programs*, 16 ELEC. BUS. 49 (Feb. 5, 1990). The U.S. Supreme Court in *Kewanee Oil Co. v. Bicron Corp.* defined reverse engineering as the act of working backwards from a completed work to develop a new work. *See* Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 476 (1974); *see also* N.Y. City Bar Ass'n Committee on Computer Law, *Reverse Engineering and Intell. Prop. Law*, 44 THE RECORD 132 (1989) [hereinafter *Reverse Engineering*]; Staines, *An Assessment of the European Commission's Proposal for a Council Directive on the Legal Protection of Computer Programs*, COMPUTER LAW. 19, 19-20 (Sept. 1989).

[9] Kellaway, *Unhappy Interface for Computer Industry Rivals*, Fin. Times, Feb. 15, 1990, § I, at 7. The European Committee for Interoperable Systems ("ECIS") "argues that in explicitly banning reverse engineering, the directive is preventing the industry from a legitimate way of understanding how the interface works. . . . The other side is the Software Action Group for Europe (SAGE), which . . . . [argues that to] allow reverse engineering would simply be an invitation to the pirates." *Id.*; *see infra* notes 102-36 and accompanying text (discussing opposing viewpoints on reverse engineering).

[10] *See infra* notes 119-36 and accompanying text (discussing arguments against reverse engineering).

[11] Directive, *supra* note 2, art. 6, at 45. Article 6 provides that

> 1. [t]he authorization of the rightholder shall not be required where reproduction of the code and translation of its form within the meaning of Article 4(a) and (b) are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided that the following conditions are met: (a) these acts are performed by the licensee or by another person having a right to use a copy of a program, or on their behalf by a person authorized to do so; (b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in subparagraph (a); and (c) these acts are confined to the parts of the original program which are necessary to achieve interoperability.
>
> 2. The provisions of paragraph 1 shall not permit the information obtained through its application: (a) to be used for goals other than to achieve interoperability of the independently created computer program; (b) to be given to others, except when necessary for the interoperability of the independently created computer program; or (c) to be used for the develop-

U.S. law is unclear.[12] Reverse engineering is not explicitly covered by U.S. statutory law.[13] U.S. courts, moreover, have not taken a definitive stand on reverse engineering in computer software cases.[14]

This Note compares the reverse engineering exception in the Directive with U.S. copyright law. Part I sets forth the background of copyright law for computer software, with emphasis on the act of reverse engineering. Part II charts the evolution of the reverse engineering exception in the Directive, and the debate that this exception has spurred. Part III compares the Directive's reverse engineering exception with the U.S. copyright law that is applicable to reverse engineering. Part IV argues that U.S. law prohibits reverse engineering, and that the Community could better serve the European software industry by similarly granting broad copyright protection without a reverse engineering exception. This Note concludes that the inclusion in the Directive of an exception permitting unauthorized reverse engineering is misguided. Instead, Community legislation should confer broad copyright protection to the rightholder.

I. *COMPUTER SOFTWARE IN THE COPYRIGHT CONTEXT*

A computer program is a detailed sequence of instructions that, when executed by a computer, brings about a desired re-

> ment, production or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyright.
>
> 3. In accordance with the provisions of the Berne Convention for the protection of Literary and Artistic Works, the provisions of this Article may not be interpreted in such a way as to allow its application to be used in a manner which unreasonably prejudices the rightholder's legitimate interests or conflicts with a normal exploitation of the computer program.

Id.

[12] *See Reverse Engineering*, *supra* note 8, at 136-37; Grogan, *Decompilation and Disassembly: Undoing Software Protection*, COMPUTER LAW. 1 (1984); Note from Thomas Niles, U.S. Ambassador to the European Community, at 2 (Jan. 24, 1990) (copy on file at the *Fordham International Law Journal* office) (discussing treatment of computer software under U.S. copyright law) [hereinafter U.S. Ambassador Note]. Ambassador Niles stated that "[u]nder U.S. law, unauthorized reproduction or translation of substantial parts of a program's object code into source code would result in a prima facie violation of the copyright owner's rights to reproduce the work and to make derivative works, granted in section 106(1) and (2)." *Id.*

[13] *See* U.S. Ambassador Note, *supra* note 12, at 2.

[14] *See infra* notes 203-37 (discussing computer software cases on reverse engineering).

sult.[15] An author writes a program in an easily-read format called source code.[16] The author then compiles the source code into object code, which can be read by the computer.[17]

The cost of researching and developing computer programs is greater than the cost of duplicating the program.[18] The software industry is therefore vulnerable to pirate activities.[19] Given the extensive investment of time and money necessary to develop a program, a lack of legal protection against piracy may deter authors from expending their resources to create new software.[20] The United States and the Community, therefore, have found it essential to create a legal environment that affords authors of software a degree of protection comparable to that given to other intellectual property.[21] The United

[15] *See* 17 U.S.C. § 101 (1988); *see also* R. NIMMER, *supra* note 1, at 12. Section 101 defines a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (1988).

[16] *See* R. NIMMER, *supra* note 1, at 13.

[17] *See id.* at 14. The success of a program depends upon the skills of the programmer. *Id.* The speed, efficiency, and style of a computer program hinges upon how a programmer chooses to structure the program's instructions. *Id.* Elements of creativity, skill, and inventiveness manifest themselves in the structure of the program. *Id.*

[18] *See id.* at 9; CONTU REPORT, *supra* note 1, at 10 (noting that computer programs are result of great intellectual effort and yet are relatively simple to duplicate).

[19] CONTU REPORT, *supra* note 1, at 10. In recent years, programs have been stored on magnetic media tapes and discs that have revolutionized the ease of storing programs, but at the same time have made verbatim copying effortless. *See* Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine Readable Form*, 1984 DUKE L.J. 663, 689-90. The software industry calculates that it loses US$3 billion a year to illegal copies in the United States and US$5.3 billion a year in Europe. *See* Bulkeley, *Software Makers are Pursuing Pirates Around the Globe with Fleets of Lawyers*, Wall St. J., Dec. 13, 1990, at B1, col. 3.

[20] *See* CONTU REPORT, *supra* note 1, at 10. The CONTU Commission thus stated that

> the following proposition seems sound: if the cost of duplicating information is small, then it is simple for a less than scrupulous person to duplicate it. This means that legal as well as physical protection for the information is a necessary incentive if such information is to be created and disseminated.

Id. The CONTU Commission further concluded that "some form of protection is necessary to encourage the creation and broad distribution of computer programs in a competitive market." *Id.* at 11; *see* EUROPEAN PARLIAMENT COMMITTEE ON ECONOMIC AND MONETARY AFFAIRS AND INDUSTRIAL POLICY, DRAFT OPINION 1 (Sept. 20, 1989) (copy on file at the *Fordham International Law Journal* office).

[21] *See* 17 U.S.C. §§ 101-17 (1988); Directive, *supra* note 2, art. 1, at 44; CONTU REPORT, *supra* note 1, at 11; *supra* note 2 (discussing U.S. and EEC recognition of copyright as legal protection for computer programs).

States and the Community favor copyright law as the best protection for computer software.[22]

A. *Copyright Law and Its Applicability to Computer Software*

Copyright laws protect original works of authorship fixed in a tangible medium.[23] Copyright owners have exclusive rights to reproduce and sell their copyrighted work.[24] In the United States, a copyright owner also has the exclusive right to create derivative works based on the original work.[25] Copyright law prevents authors from protecting their underlying ideas, but retains the author's exclusivity over their expression.[26] This idea/expression dichotomy benefits the public because the complete access to ideas allows other authors to express the idea in a different manner and distribute their expression to the public.[27] The idea/expression dichotomy also benefits the holder of the copyright because the unique manner of expressing the idea may not be marketed by another.[28] The United States and the Community apply these copyright

[22] *See* Directive, *supra* note 2, art. 1, at 44; *see also* CONTU REPORT, *supra* note 1, at 16-17 (discussing benefits of copyright protection over other forms of legal protection for computer programs, such as patent and trade secret).

[23] *See* 17 U.S.C. § 102(a) (1988); Directive, *supra* note 2, art. 1, at 44; R. NIMMER, *supra* note 1, at 4. Section 102(a) provides that "[c]opyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression, now known or later developed." 17 U.S.C. § 102(a) (1988). Section 101 defines a fixed work as being "in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration." *Id.* § 101.

[24] *See* 17 U.S.C. § 106 (1988); *supra* note 7 (containing text of section 106); *see also* Directive, *supra* note 2, art. 4, at 44; *infra* note 79 (containing text of article 4).

[25] 17 U.S.C. § 106(2) (1988); *see supra* note 7 (containing text of section 106). Section 101 defines a derivative work as

> a work based upon one or more preexisting works, such as a translation . . . or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a 'derivative work'.

17 U.S.C. § 101 (1988).

[26] *See* 17 U.S.C. § 102(b) (1988); Directive, *supra* note 2, art. 1(2), at 44. The U.S. statute provides that "[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." 17 U.S.C. § 102(b) (1988).

[27] *See* R. NIMMER, *supra* note 1, at 5.

[28] *Id.*

principles to computer software for several reasons.[29]

Computer programs are classified as literary works for the purposes of copyright protection.[30] This classification is advantageous to programmers because the application of copyright principles to literary works is well established.[31] Most copyright law that is applicable to literary works can also be applied to computer programs.[32] The application of these copyright principles to computer programs promotes familiarity, which, in turn, makes the development of new products easier because software authors readily recognize what may and may not be protected by copyright.[33]

Moreover, several countries recognize copyright as the legal protection for computer software.[34] Although the Berne

[29] *See infra* notes 30-42 and accompanying text (discussing advantages of applying copyright law to computer programs).

[30] *See* Directive, *supra* note 2, art. 1(1), at 44; CONTU REPORT, *supra* note 1, at 16.

[31] BUSINESS SOFTWARE ALLIANCE (BSA), WHITE PAPER: MODEL GUIDELINES FOR THE LEGAL PROTECTION OF COMPUTER PROGRAMS 1-2 (Jan. 1990) (copy on file at the *Fordham International Law Journal* office) [hereinafter BSA WHITE PAPER]. BSA advocates that classifying computer programs as literary works is advantageous because

> there exists a well-developed body of copyright legal principles applicable to literary works. This includes not only the rules developed under national laws, but also under the Berne Convention for the Protection of Literary and Artistic Works, which provides a well-developed system of rules applied throughout the world on the basis of the member nations' reciprocal recognition of copyright protection.

Id. BSA is an interest group that represents U.S. and international software producers. *Id.*

[32] *See* CONTU REPORT, *supra* note 1, at 16. In applying copyright law to computer programs, the CONTU Commission found that the law required no changes in order to afford copyright protection to programs. *Id.*

[33] BSA WHITE PAPER, *supra* note 31, at 2. BSA advocates that

> [b]y applying these [copyright principles] to computer programs, governments can promote legal certainty that in turn encourages software companies to develop new products. . . . This consistency encourages a country's trade and technological development by making it easier for foreign software companies to bring their existing products into a market without the risk associated with new and untested legal rules. Equally important, it encourages domestic authors to create new computer programs with the assurance that their products will be protected under well-developed and consistent rules, both at home and abroad.

Id.

[34] Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 6. The Commission in the Proposed Directive noted that

> [t]he following countries have explicitly recognized the protection of computer programs by copyright: Australia, Brazil, Chile, Dominican Republic,

Convention[35] and the Universal Copyright Convention[36] do not specifically protect computer software, it is generally acknowledged that if a new work has the requisite level of creativity, it will be protected under these international copyright conventions.[37] The international familiarity with copyright benefits software authors who market their programs in more than one country.[38]

Furthermore, protecting computer software by copyright guards against the rise of monopolies in the computer software industry.[39] As noted, copyright protects the expression but not the underlying idea of a work.[40] The rationale behind this

> France, Federal Republic of Germany, Hungary, India, Indonesia, Japan, Malaysia, Mexico, Philippines, Republic of Korea, Singapore, Spain, Trinidad and Tobago, Turkey, United Kingdom, United States of America. Draft laws are also under consideration in a number of countries to the same effect, including Denmark, Italy and the Netherlands.

Id.

[35] Sept. 9, 1886, 12 Martens Nouveau Recueil (ser. 2), 173, 168 Parry's T.S. 185, *entered into force for the United States* Mar. 1, 1989.

[36] Sept. 6, 1952, 6 U.S.T. 2731, T.I.A.S. No. 3324, 216 U.N.T.S. 132, *amended* July 24, 1971.

[37] *See* Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 8. The Commission stated that

> [c]opyright has the added advantage of affording a high level of international protection to works so covered, through the application of the Berne and Universal Copyright Conventions. Although neither convention expressly mentions computer programs among the works to be covered by copyright it is generally understood that as new forms of intellectual property are developed they will be encompassed by the conventions in so far as the same kinds of creativity are involved in the elaboration of such new forms of work as for existing works.

Id. The U.S. Supreme Court has stated that copyright law requires that a work "[possess] at least some minimal degree of creativity. . . . To be sure, the requisite level of creativity is extremely low; even a slight amount will suffice." Feist Publications, Inc. v. Rural Tel. Serv. Co., Inc., 111 S. Ct. 1282, 1287 (1991).

[38] *See supra* notes 34-37 and accompanying text (discussing international acceptance of copyright law and its significance for computer programs).

[39] *See* Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 7; C. SHERMAN, H. SANDISON & M. GUREN, COMPUTER SOFTWARE PROTECTION LAW 204:23-24 (1989). The Commission favored copyright because "[c]opyright protection does not grant monopolies hindering independent development. Copyright protects only the expression but not the underlying idea of a work. It does not therefore block technical progress or deprive persons who independently developed a computer program from enjoying the benefits of their labour and investments." Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 7; *see* R. NIMMER, *supra* note 1, at 6.

[40] *See* R. NIMMER, *supra* note 1, at 5. Copyright protection attaches only to forms of expression, not to the ideas expressed or processes described. *Id.* The U.S. Supreme Court stated that the

idea/expression dichotomy is that ideas should not be withdrawn from use by the public because doing so would inhibit rather than promote creative advancement.[41] Large companies, therefore, must share their ideas rather than monopolize them.[42]

Problems arise, however, in the application of the idea/expression dichotomy to computer software.[43] The actual object and source codes are generally classified as protected expression.[44] One problem, however is whether the ideas embodied in the object and source codes are to be treated as protected expressions or as unprotected ideas.[45] If the ideas within the codes are unprotected, copyright law does not pro-

> idea/expression . . . dichotomy, applies to all works of authorship. As applied to a factual compilation, assuming the absence of original written expression, only the compiler's selection and arrangement may be protected; the raw facts may be copied at will. This result is neither unfair nor unfortunate. It is the means by which copyright advances the progress of science and art.

Feist Publications, 111 S. Ct. at 1290; *see* Mazer v. Stein, 347 U.S. 201 (1954); Baker v. Selden, 101 U.S. 99 (1879); Stern Elec., Inc. v. Kaufman, 669 F.2d 852 (2d Cir. 1982); Morrisey v. Proctor & Gamble Co., 379 F.2d 675 (1st Cir. 1967).

[41] *See* R. NIMMER, *supra* note 1, at 5. Professor Nimmer states that "the distinction between expression and ideas or processes defines a system that encourages creative expression while safeguarding the right of others to use, adapt, and build on an idea." *Id.*

[42] Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 7.

[43] *See* CONTU REPORT, *supra* note 1, at 18; R. NIMMER, *supra* note 1, at 5. The CONTU Commission stated that "the distinction between copyrightable computer programs and uncopyrightable processes or methods of operation does not always seem to 'shimmer with clarity.'" CONTU REPORT, *supra* note 1, at 18. Professor Nimmer states that

> [t]he most controversial criterion for copyrightability of computer software is the Section 102 exclusion of any copyright protection for ideas . . . . In print or other traditional forms of communication, this distinction is relatively easy to describe. The plot of a novel is not copyrightable, but the dialogue written by an author is protected.

R. NIMMER, *supra* note 1, at 5.

[44] *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1239 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1246-47 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 857 (2d. Cir. 1982); Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1498 n.11 (D. Minn. 1985).

[45] *See* C. SHERMAN, H. SANDISON & M. GUREN, *supra* note 39, at 204:24. This book emphasizes that "it is not so easy to determine which elements of the program itself are protectable expressions and which elements are unprotectable ideas." *Id.*

hibit the public from accessing those ideas.[46] Some commentators assert, however, that if the code is classified as an unprotected idea, then copyright law will not adequately protect computer software because the idea is often the innovative element that requires protection.[47] If anyone can access the idea elements of the code, these commentators assert, a program can be duplicated easily and copyright law will grant minimal protection against piracy.[48] Classifying the codes as ideas, therefore, does not adequately protect the copyright holder from piracy.[49]

Alternatively, classifying the underlying principles in the codes as expression would prevent the public from utilizing such principles.[50] Without specific limitations, however, this expanded protection could create monopolies in the computer industry.[51] The debate about protecting source and object code relates directly to reverse engineering, a process that has

[46] *See* R. NIMMER, *supra* note 1, at 5.

[47] *See* Samuelson, *supra* note 19, at 754-55; *see also* AMERICAN SOCIETY FOR INFORMATION SCIENCE, OMNIBUS COPYRIGHT REVISION 97 (1973) [hereinafter OMNIBUS COPYRIGHT REVISION]; Goldberg, *Legal Protection of EDP Software*, 18 DATAMATION 66, 67 (1972); Stern, *The Case of the Purloined Object Code: Can It Be Solved? Part 2*, BYTE, Oct. 1982, at 210 & 214. Professor Samuelson stated that

> [p]erhaps an even stronger reason for dissatisfaction is that copyright law does not aim to protect that which sophisticated programmers and those who market the programs think are the most commercially valuable aspects of the programs: either the algorithm of the program, the elaborate logical structure of the program, or some "trick" that makes the program operate faster or more efficiently than others. What the programmer wants to be compensated for is the value of these ideas, not just for the particular words or symbols used to express them.

Samuelson, *supra* note 19, at 755.

[48] *See* Samuelson, *supra* note 19, at 755-56; *see also* OMNIBUS COPYRIGHT PROVISION, *supra* note 47, at 97; Goldberg, *supra* note 47, at 67; Stern, *supra* note 47, at 214.

[49] *See* Samuelson, *supra* note 19, at 755; *see also* OMNIBUS COPYRIGHT REVISION, *supra* note 47, at 97; Goldberg, *supra* note 47, at 67; Stern, *supra* note 47, at 214.

[50] *See* R. NIMMER, *supra* note 1, at 9. Professor Nimmer stated that "[p]rotection for expression and authorship is significant, but extends only insofar as it can be achieved without creating a monopoly on ideas, processes, or objects." *Id.*

[51] *See* C. SHERMAN, H. SANDISON & M. GUREN, *supra* note 39, at 204:25 (citing Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987)). The authors of this book stated that "[t]he idea/expression test set forth in *Whelan* appears to be extremely broad, and it is not yet clear how far other courts will be willing to embrace it." *See id.* The main problem with the test is that the *Whelan* definition of protectable expression may be so broad that virtually no aspect of a computer program is excluded. *See id.*

stirred controversy in the context of granting copyright protection to computer software.

B. *Reverse Engineering*

Reverse engineering involves creating a new program based on information obtained from the original program.[52] The author of a computer program writes source code that can be compiled into a program.[53] Prior to the program's distribution, the author compiles the source code into object code that can be read by the computer, but is indecipherable to the user of the program.[54] When a program is publicly distributed, it consists of the object code—the source code is guarded by the programmer.[55] A reverse engineer works backward from a computer's object code to uncover the structure and organization of the source code.[56]

The act of reverse engineering usually entails the decompiling of a computer program's object code into source code.[57] The source code is then studied to ascertain how the program works.[58] The knowledge thus obtained may be used for many commercial and noncommercial purposes, including research or the creation of a competing product.[59]

[52] *See Reverse Engineering, supra* note 8, at 132.

[53] *See* R. NIMMER, *supra* note 1, at 12-13.

[54] *See id.*

[55] *See Reverse Engineering, supra* note 8, at 133. The Computer Law Committee of the New York City Bar Association (the "Computer Law Committee") explained that "the source code—which could reveal to any trained programmer who examined it the structure and organization of the program as well as its logical techniques (called algorithms)—is almost always treated by the developer as a closely guarded trade secret." *Id.*

[56] *See id.*

[57] *See id.*

[58] *See id.* at 133-35.

[59] *See id.* According to the Computer Law Committee, reverse engineering has both commercial and non-commercial uses such as:

> 1. Reverse engineering can be conducted in a primarily non-commercial context, to learn about the structure and organization of the product or to learn its algorithms: (a) As part of a classroom exercise; or (b) In connection with the writing of a textbook . . . .
> 2. Reverse engineering can be conducted in a commercial setting where the goal is not to learn the entire structure and organization of the program and its algorithms but to learn enough about the program: (a) To develop a non-competing program which interacts with the first program . . . ; (b) To develop a program which interacts with the first program and changes it in some way contrary to the wishes of its developer . . . ; (c) To develop a

One of the purposes of reverse engineering is achieving interoperability.[60] Interoperability allows one program to link into another program.[61] The Council of Ministers of the European Communities (the "Council") and many small software manufacturers assert that interoperability is the primary purpose for decompiling and reverse engineering.[62] According to small software manufacturers, in order to obtain the information needed to create interoperable programs, it is necessary to decompile and reverse engineer from the original program.[63]

> competing, generally independently created program that "understands" the file and command structure of the first program and can therefore operate on files or documents created using the first program . . . .
>
> 3. Reverse engineering can be conducted in a commercial setting where the goal is to learn the entire structure and organization of the first program and all of its algorithms: (a) To develop a directly competing program functionally interchangeable program which uses only unprotected features of the first, that is, ideas which are not subject to copyright and have not been patented; (b) To develop a directly competing program which may or may not be functionally interchangeable with the first but which uses protected features of the first . . . .

Id.

[60] *See id.*

[61] Directive, *supra* note 2, at 43. The Council defines interoperability as the "functional interconnection and interaction [between programs] . . . . [S]uch interoperability can be defined as the ability to exchange information and mutually to use the information which has been exchanged." *Id.*

Scholars have noted that

> [s]oftware compatibility is obviously an important commercial objective in the computer industry. When one manufacturer creates a best-selling computer utilizing a unique operating system, its competitors will want to develop compatible operating systems for their own computers to enable buyers of the second and subsequent machines to take advantage of all the application programs written for the first machine. Similarly when one company develops a best-selling application program, its competitors will want to develop compatible programs to allow users familiar with the first program to use the new program without reformatting their data and without retraining their personnel.

C. SHERMAN, H. SANDISON & M. GUREN, *supra* note 39, at 210:42.

[62] *See* Directive, *supra* note 2, at 43; *Defense Against Pirates or Death to the Clones?*, BUS. WK. 138, 140 (May 7, 1990) [hereinafter *Defense Against Pirates*]. The Council stated that "circumstances may exist when such a reproduction of the code and translation of its form . . . are indispensable to obtain the necessary information to achieve the interoperability of an independently created program with other programs." Directive, *supra* note 2, at 43.

[63] *See Defense Against Pirates*, *supra* note 62, at 140; *Pieces of 1-2-3*, THE ECONOMIST, July 14, 1990, at 73. The article in the *Economist* noted that

> [i]n order to build "interoperable" software packages (packages that can run on a variety of computers in concert with a variety of programs), program-

The extent of legal protection for computer software is not clear in situations involving reverse engineering.[64] Under the U.S. Copyright Act, the copyright owner is the only party authorized to create a work based on the original work.[65] Unauthorized reverse engineering to create an interoperable program may therefore infringe upon the owner's exclusive right to create derivative works.[66]

In the reverse engineering context, however, the exclusive rights to copy and to create derivative works may be at odds with the idea/expression dichotomy that mandates public access to codes if codes are classified as ideas.[67] Neither U.S. nor EEC copyright law protects ideas, processes, and procedures.[68] Arguably, no legal prohibition therefore exists against accessing the idea by decompiling the object code and reverse engineering a new program.[69] The institutions of both the United States and the Community have struggled with the issue of whether reverse engineering is a permitted exception to copyright protection.[70]

II. *REVERSE ENGINEERING IN THE EUROPEAN COMMUNITY*

In October 1990, the Commission of the European Com-

> mers need to take other companies' programs apart to see how they work. . . . Companies with no established proprietary systems software of their own . . . complain that unless the directive allows reverse engineering, they will have difficulty in developing new applications software—since their programs draw on their competitors' systems products.

Id.

[64] *See* D.C. Toedt III, The Law and Business of Computer Software 2:55 (1990). Professor Toedt states that "it is difficult to advise a client about the propriety of reverse- engineering a competitor's computer software." *Id.*

[65] 17 U.S.C. § 106(2) (1988); *see supra* note 7 (containing text of section 106).

[66] *See* U.S. Ambassador Note, *supra* note 12, at 2.

[67] *See id.* at 2-3. Ambassador Niles stated that "[u]nder U.S. law, a party accused of violating the protection granted by the copyright law could raise a number of potential defenses. These include . . . attacking the copyrightability of the allegedly infringed program, arguing that copying was limited to unprotected elements of the program (e.g., ideas or expression inextricably merged with such ideas)." *Id.*

[68] *See* 17 U.S.C. § 102(b) (1988); *supra* note 26 (containing text of section 102(b)). Like the United States, the Community also does not protect ideas. Directive, *supra* note 2, art. 1(2), at 44.

[69] *See* 17 U.S.C. § 102(b) (1988); Directive, *supra* note 2, art. 1, at 44.

[70] *See generally* U.S. Ambassador Note, *supra* note 12; Colombe & Meyer, *Interoperability Still Threatened by E.C. Software Directive: A Status Report*, Eur. Intell. Prop. Rep., 325, 325-26 (Sept. 1990).

munities (the "Commission") issued the Proposed Directive to harmonize the laws of the Member States and to create a uniform system of legal protection for computer software.[71] The Council, in May 1991, enacted this Directive that permits reverse engineering without authorization from the copyright owner for the purpose of creating interoperable programs.[72] This reverse engineering provision has caused controversy among international software manufacturers.[73]

A. *General Provisions in the Directive*

The Directive recommends that computer software be afforded the same protection as literary works, and that copyright be the means of protection.[74] Under the Directive, a computer program is protected as expression.[75] To qualify for copyright protection under the Directive, a program must be an original creation by the author.[76]

The Directive broadly prohibits acts of wilful copying and circulation for commercial purposes, and requires Member States to take action against such activities.[77] The term of protection granted by the Directive is for the life of the author

[71] *See* Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 5. "The current absence of such clear and congruent legislative provisions in Member States concerning the rights of authors of computer programs has thus prompted the Commission to make this proposal to the Council." *Id.*

[72] *See* Directive, *supra* note 2, art. 6, at 45; *supra* note 11 (containing text of article 6).

[73] *See infra* notes 102-36 and accompanying text (discussing debate on reverse engineering provision in Proposed Directive).

[74] *See* Directive, *supra* note 2, art. 1, at 44.

[75] *See id.* art. 1(2), at 44 (stating that "[i]deas and principles [that] underlie any element of [the] program . . . are not protected").

[76] *See id.* art. 1(3), at 44.

[77] *See id.* art. 7, at 45. Article 7 of the Directive states that

> Member States shall provide, in accordance with their national legislation, appropriate remedies against a person committing any of the acts listed in subparagraph (a), (b), and (c) below: (a) any act of putting into circulation of a copy of a computer program knowing, or having reason to believe that it is an infringing copy; (b) the possession, for commercial purposes, of a copy of a computer program knowing, or having reason to believe, that it is an infringing copy; (c) any act of putting into circulation, or the possession for commercial purposes of, any means the sole intended purpose of which is to facilitate the unauthorized removal or circumvention of any technical device which may have been applied to protect a computer program.

Id.

plus fifty years.[78]

The Directive grants the copyright owner exclusive rights to distribute, reproduce, translate, adapt, arrange, and alter the computer program.[79] Others may partake in these exclusive rights only with authorization from the copyright owner.[80] The Directive, however, permits unauthorized use of the program to make a back-up copy.[81] The Directive also permits a user to study the program's ideas and principles while performing the acts of loading, displaying, running, transmitting, or storing the program.[82]

A controversial exception to the author's exclusive rights under the Directive is the right to reverse engineer to achieve interoperability.[83] The original draft of the Directive did not include a reverse engineering exception, and it was only after

[78] *See id.* art. 8, at 45. Article 8 of the Directive states that "[p]rotection shall be granted for the life of the author and for fifty years after his death." *Id.*

[79] *See id.* art. 4, at 44. Article 4 grants the rightholder the exclusive right to do or to authorize:

> (a) the permanent or temporary reproduction of a computer program by any means and in any form, in part or in whole. Insofar as loading, displaying, running, transmission or storage of the computer program necessitate such reproduction, such acts shall be subject to authorization by the rightholder; (b) the translation, adaptation, arrangement and any other alteration of a computer program and the reproduction of the results thereof, without prejudice to the rights of the person who alters the program; (c) any form of distribution to the public.

Id.

[80] *See id.*

[81] *See id.* art. 5, at 44-45. Article 5 grants the following exceptions to the restricted acts listed in article 4:

> 1. In the absence of specific contractual provisions, the acts referred to in Article 4(a) and (b) shall not require authorization by the rightholder where they are necessary for the use of the computer program by the lawful acquiror in accordance with its intended purpose, including for error correction.
>
> 2. The making of a back-up copy by a person having a right to use the computer program may not be prevented by contract insofar as it is necessary for that use.
>
> 3. The person having a right to use a copy of a computer program shall be entitled, without the authorization of the rightholder, to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he is entitled to do.

Id.

[82] *Id.* art. 5(3), at 45.

[83] *See id.* art. 6, at 45; *supra* note 11 (containing text of article 6); *infra* notes

considerable debate that the Parliament of the European Communities (the "Parliament") included such a provision.[84]

B. *The Reverse Engineering Exception*

1. The General Provisions of Article 6

Article 6 of the Directive permits the decompiling of object code to achieve interoperability.[85] Furthermore, article 6 permits unauthorized reproduction and translation to obtain the information necessary to achieve interoperability.[86] The article requires, however, that only when the information necessary to achieve interoperability is unpublished or unavailable may a third party decompile the program.[87] Additionally, only those parts of the original program that are necessary to achieve interoperability may be decompiled.[88]

Article 6 does prohibit all circulation of information obtained through decompiling except for the purpose of creating an interoperable program.[89] To combat misuse of reverse engineering, the article also contains a safety clause that prohibits the information from being used for any purpose that would infringe upon the copyright of the original program.[90]

102-36 (discussing controversy that arose from Directive's reverse engineering amendment).

[84] *See* Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 16. The Commission stated that "the ability of a competing manufacturer to write an independent but compatible program often depends on his possibility to have access to the target program or to certain information relating to it. Access to information is not a matter of copyright law." *Id.* In the explanatory statement the Parliament

> decided to open up [the area of decompiling], with the clear purpose of allowing the interoperability of systems and thus helping to crete [sic] more computer programs and increase the creativity of programs produced by either individuals or groups of persons, or businesses, especially small or medium-sized ones, without forgetting the growing needs and difficulties of users and whilst preventing the software market from becoming totally dominated by powerful hardware companies.

PARLIAMENT AMENDMENTS, *supra* note 3, at 7 (explanatory statement).

[85] Directive, *supra* note 2, art. 6, at 45; *see supra* note 11 (containing text of article 6).

[86] *See* Directive, *supra* note 2, art. 6, at 45.

[87] *See id.*

[88] *See id.*

[89] *See id.*

[90] *See id.* art. 6(2)c, at 45.

2. Background and Policy Considerations

The original draft of the Directive did not seek to resolve the interoperability issue.[91] Rather, it left open whether a user could decompile and reverse engineer to achieve interoperability.[92] The Commission recognized that an interoperability provision could prevent the rise of monopolies, but it believed that such a provision was unnecessary because EEC competition law already protected against monopolies.[93]

The Parliament, however, feared that without a provision permitting decompiling for interoperability purposes, small European software authors would be at a competitive disadvantage with larger U.S. manufacturers.[94] Parliament thus introduced articles 5(a) and 5(2a), new provisions that allow decompiling and reverse engineering to achieve interoperability.[95]

[91] *See* Commission's Proposed Directive, *supra* note 3, O.J. C 91/4; *supra* note 84 (discussing Commission's deliberate intention not to address issue of interoperability); *Parliament Okays Software Directive with Reverse Engineering Compromise*, 4 WORLD INTELL. PROP. REP. 176 (Aug. 1990) [hereinafter *Parliament Okays Software Directive*].

[92] *Parliament Okays Software Directive*, *supra* note 91, at 177.

[93] Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 16. In the conclusions following the text of the Proposed Directive, the Commission suggested that

> [a]ny arrangement or measure which goes beyond the existence of copyright can be subject to control under the competition rules. This means that for example any attempt to extend by contractual agreements or other arrangements the scope of protection to aspects of the programs for which protection under copyright is not available, or the prohibition of any act which is not reserved for the right owner may constitute an infringement of the competition rules.
>
> Moreover, companies in a dominant position must not abuse that position within the meaning of Article 86 of the Treaty. For example, under certain circumstances the exercise of copyright as to the aspects of a program, which other companies need to use in order to write compatible programs, could amount to such an abuse. . . .
>
> Furthermore, the ability of a competing manufacturer to write an independent but compatible program often depends on his possibility to have access to the target program or to certain information relating to it. Access to information is not a matter of copyright law. Article 86 always applies where a dominant company abusively refuses access to such information or restricts unreasonably such access.

Id.

[94] *See* PARLIAMENT AMENDMENTS, *supra* note 3, at 7 (explanatory statement).

[95] *Id.* arts. 5(a) & 5(2a), at 33-34. Article 5(a) of the Parliament's amendments provided that

> [n]otwithstanding any contractual arrangements to the contrary, the rights

Upon reading Parliament's recommendations, the Commission issued an Amended Proposed Directive that included a reverse engineering exception.[96] The Amended Proposed Di-

> enumerated in Article 4(a) and (b) shall not be exercised by the author to prevent any act essential to ensure the maintenance of the program and the creation or operation of interoperable programs. This option may only be exercised by the licensee or by another person entitled to use a copy of the program on his behalf by the person authorized to do so and only where the following conditions are fulfilled: (a) the information necessary to achieve interoperability shall not have been published or made available previously; (b) the retrieval of information shall be confined to the parts of the original program which are necessary for the achievement of this aim; (c) the information retrieved may not be communicated to third parties except in so far as this is necessary for the operation of the second program; (d) the information retrieved may not be used to create or market a program, which violates a copyright or the program of origin.
>
> The provisions of this article may not be interpreted in such a way as to allow information obtained in the application thereof to be used in a manner which unjustifiably damages the legitimate interests of the right-holder or which is contrary to the normal operation of the program.

Id. Parliament's article 5(2a) provided that

> [n]otwithstanding the provisions of Article 4(a), the legitimate owner of a copy of a program may, without having to request the authorization from the right-holder, observe, study or test the working program in order to determine its underlying ideas, principles and other characteristics where these are not protected by copyright, in the course of loading, viewing, running, transmission or storage in the execution of his contractual duties.

Id.

[96] *See* Amended Proposed Directive, *supra* note 3, art. 5 *bis*, at 28-29. The Commission's amended article 5 *bis* provides that

> 1. [n]otwithstanding contractual provisions to the contrary, the authorization of the owner of the rights shall not be required where reproduction of the code and translation of its form are indispensable to achieve the creation, maintenance or functioning of an independently created interoperable program, provided that the following conditions are met: a) these acts are performed by the licensee or by another person having a right to use a copy of a program, or on their behalf by a person authorized to do so; b) the information necessary to achieve interoperability has not previously been published, or made available to the persons referred to in subparagraph a); and c) these acts are confined to the parts of the original program which are necessary to achieve interoperability with it.
>
> 2. The provisions of paragraph 1 of this Article shall not permit the information obtained through its application: a) to be used for goals other than to achieve the interoperability of the independently created program; b) to be given to others, except when necessary for the interoperability of the independently created program; or c) to be used for the creation or marketing of a program which infringes copyright in respect of the original program, and in particular of a program substantially similar in its expression.
>
> 3. In accordance with the provisions of the Berne Convention for the Protection of Literary and Artistic works, the provisions of this Article may not be interpreted in such a way as to allow its application to be used in a man-

rective included Parliament's recommendations, except that the Commission restructured Parliament's articles 5(a) and 5(2a) to restrict the situations in which the Directive would allow reverse engineering.[97] The Commission's article 5 *bis* thus reflected Parliament's recommendations as well as the widely debated concerns of software manufacturers.[98]

In a Common Position Paper, the Council subsequently proposed a new article 6 similar to the proposed article 5 *bis* that specifically addressed the decompilation issue.[99] The Council attempted to safeguard against the misuse of decompilation by providing that a program substantially similar in expression would infringe the original program's copyright.[100] Additionally, the Council referred to Community competition law which forbids a dominant supplier from refusing to make available information that is necessary for interoperability.[101]

C. *Debate on Reverse Engineering*

Two opposing viewpoints emerged as a result of the Commission's Directive.[102] The European Committee for Interoperable Systems ("ECIS") favors a provision that permits decompiling for the purpose of achieving interoperability.[103]

> ner which unreasonably prejudices the rightholder's legitimate interests or conflicts with a normal exploitation of the computer program.

Id.

[97] *Compare id. with* PARLIAMENT AMENDMENTS, *supra* note 3, arts. 5(a) & 5(2a), at 33-34. *See supra* note 95 (containing text of articles 5(a) and 5(2a)).

[98] *See* Amended Proposed Directive, *supra* note 3, at 2 (discussing Commission's reasoning for reverse engineering amendment). The Commission stated that "[i]n response to concerns expressed by the European Parliament and by part of the industry, a further exception to the author's exclusive rights for the purpose of creating an interoperable program has been accepted." *Id.*; *see infra* notes 105-18 (discussing arguments in favor of reverse engineering exception).

[99] *Compare* Directive, *supra* note 2, art. 6, at 45 *with* Amended Proposed Directive, *supra* note 3, art. 5 *bis*, at 28-29.

[100] *See* Directive, *supra* note 2, art. 6(2)c, at 45; *supra* note 11 (containing text of article 6).

[101] Directive, *supra* note 2, at 43-44. The Council recognized the Community competition laws when it emphasized in the Common Position Paper that "the provisions of this Directive are without prejudice to the application of the competition rules under Articles 85 and 86 of the Treaty if a dominant supplier refuses to make information available which is necessary for interoperability as defined in this Directive." *Id.*

[102] *See Parliament Okays Software Directive*, *supra* note 91, at 177.

[103] *Id.* at 176. ECIS represents companies such as France's Bull, Italy's Olivetti, and the United States' NCR (National Cash Register). *Id.*

The Software Action Group for Europe ("SAGE") favors restricting decompiling on the ground that it could lead to piracy.[104]

1. Arguments for Reverse Engineering

ECIS's primary argument is that prohibiting the decompiling of mass-marketed programs will render software authors unable to create new interoperable programs through reverse engineering.[105] ECIS fears that if reverse engineering were restricted, the major software makers would create a monopoly at the expense of small and medium size software manufacturers.[106] Prohibiting reverse engineering, ECIS argues, would free large manufacturers to keep their codes secret from smaller manufacturers seeking to create interoperable programs.[107] ECIS contends that an explicit decompiling provision in the Directive would force large software manufacturers to share their information with smaller manufacturers.[108]

According to ECIS and other proponents of decompiling, decompiling permits an "open" system.[109] An "open" system, they contend, enables programmers to develop programs that merge easily with other computer systems, thus ensuring inter-

[104] *See* BUSINESS SOFTWARE ALLIANCE (BSA), WHITE PAPER: COPYING THROUGH REVERSE ENGINEERING, at 8-9 (Nov. 10, 1989) (copy on file at the *Fordham International Law Journal* office) [hereinafter BSA REVERSE ENGINEERING PAPER] (BSA represents leading software producers and is a member of SAGE); 9 EUR. INTELL. PROP. REP. 325-26 (Sept. 1990). SAGE's members include the major hardware manufacturers such as IBM and DEC. *Id.*

[105] *EC Debates Proposal on Interfaces and Reverse Engineering*, 16 Computer L. & Tax Rep., 4-5 (Mar. 1990). Companies such as "Unisys, NCR, Amdahl and Fujitsu . . . view their economic viability as dependent upon the availability on reasonable terms of sufficiently detailed specifications to achieve inter-operability." *Id.* at 5.

[106] *See* Bernier, *supra* note 8, at 49. Ms. Bernier quotes Mr. Philippe Wacker, a Brussels lobbyist representing ECIS, who advocates that " '[e]veryone does reverse engineering, not to copy programs, but to survive' . . . . He explains that companies analyze software to develop interoperable and new products. Specifically restricting reverse analysis, he says, would reinforce the dominant position of the major software makers and impede the growth of small and medium-sized suppliers." *Id.*

[107] *See id.*

[108] *See id.* Proponents of reverse engineering are primarily small and medium size companies that want to research the technology of the larger companies. Computer L. & Tax Rep., *supra* note 105, at 5.

[109] *See EC Parliament Approves Software Directive that Includes Reverse Engineering Provision*, 7 Int'l Trade Rep. (BNA) 1165 (July 25, 1990) [hereinafter Trade Reporter].

operability.[110] Prohibiting decompiling, on the other hand, prevents people from researching and analyzing computer programs to achieve interoperability.[111] ECIS concludes that if programmers are unable to reverse engineer to create interoperable programs, the system "closes."[112]

ECIS also argues that permitting decompiling is not illegal under copyright law because copyright law does not protect ideas.[113] ECIS notes that decompiling attempts to decipher the underlying source and object code of the computer program.[114] The principles embodied in the codes, argues ECIS, are not protected expressions but rather are part of the underlying idea that one can use without infringing a copyright.[115] ECIS states that because some of the principles embodied in the codes are ideas, the public should have complete access to the codes through decompiling.[116] ECIS thus concludes that copyright law does not prohibit decompiling of a computer program.[117]

The ECIS campaign succeeded because the Council adopted the Directive with a provision that permits decompiling for the purpose of creating interoperable programs through reverse engineering.[118]

[110] *See* Fin. Times, Sept. 11, 1990, § I, at 20.

[111] *See* Bernier, *supra* note 8, at 49.

[112] *See id.*

[113] *See Defense Against Pirates*, *supra* note 62, at 138, 140. The article stated that

> ECIS says it is asking for nothing more than what computer makers already have. Software copyrights protect only lists of program instructions. The law does not, ECIS argues, protect the ideas embodied in those instructions, such as the sequence of interface messages. In the U.S., no law expressly prohibits reverse engineering, and companies routinely dissect one another's programs to get at those ideas. Says a lawyer at one U.S.-based ECIS member company: 'It's intellectually inconceivable that copyright law can prevent people from reading a program.'

Id.

[114] *See id.*

[115] *Id.*

[116] *Id.*

[117] *Id.*

[118] *See* Directive, *supra* note 2, art. 6, at 45; *see also* Trade Reporter, *supra* note 109. It was only after "the formation of the lobby group ECIS . . . which favors so-called 'open' systems, [that] the Commission agreed to revise its text to include a broad exemption for reverse engineering." *Id.*

2. Arguments Against Reverse Engineering

The opposing interest group, SAGE, argued that unlimited decompiling would result in widespread piracy.[119] SAGE was concerned by Parliament's failure to define "interoperability" in the reverse engineering provision.[120] SAGE recognized that decompiling can create interoperable programs.[121] SAGE argued, however, that if the Directive does not adequately define interoperability, decompiling would be misused and infringe the copyright.[122] SAGE recommended that the Directive define interoperability in a manner that limits the points at which a new program can attach to the original program.[123] SAGE argued that such a limitation would ensure that new programmers only had access to the parts of the original program needed to achieve interoperability.[124] The Proposed Directive's failure to define interoperability, SAGE argued, made software authors vulnerable to piracy.[125]

Several Parliament members also opposed a reverse engineering exception.[126] They argued that because of the vast amounts of time and money invested in developing software, authors need assurance that they will reap the rewards of these significant investments without the fear of piracy.[127] These Parliament members stated that forbidding decompiling would give such assurance because decompiling would be the exclusive right of the copyright owner.[128] One Parliament member

[119] *Defense Against Pirates*, *supra* note 62, at 138.

[120] *See* Colombre & Meyer, *supra* note 70, at 327.

[121] *See id.*

[122] *See id.*

[123] *See id.* at 328.

[124] *See id.*

[125] Trade Reporter, *supra* note 109, at 1166. SAGE representative in Brussels, Ms. Henriette Tielemans, noted that the failure to define interoperability provides no clear-cut safeguard against the misuse of decompilation. *Id.*

[126] *See* 1990-1991 EUR. PARL. DEB. (No. 3-392) 24, at 27-28 (July 10, 1990) (statements of representatives Lane (RDE) and Cooney (PPE) advocated prohibiting a decompiling amendment) [hereinafter PARLIAMENT DEBATES].

[127] *Id.* Representative Lane stressed a need to protect software authors because "[m]any years of intensive work is [sic] invested in . . . [software] development. Because of the size of these companies, a very large proportion of their investment is put into research and development. Their work must be protected from predators who have no investment in research and development. . . . We must ensure that protection is afforded through copyright." *Id.* at 27.

[128] *Id.* at 27-28; *see supra* note 127 and accompanying text (discussing importance of assuring authors protection from piracy through reverse engineering).

stated that restricting decompiling would not grant a monopoly to the large software houses, but rather would protect the smaller manufacturers who invest their limited resources in developing software.[129]

SAGE argued that unauthorized reverse engineering would substantially threaten the programmer's investment.[130] Reverse copiers, argued SAGE, have the ability to benefit cheaply from the software author's research and development.[131] SAGE contended that granting a broad right to engage in decompiling would force software authors to devote their resources to protecting their work, rather than to developing new products.[132]

Manufacturers who are opposed to reverse engineering point out that there are many ways to achieve interoperability without decompiling.[133] For instance, major software producers encourage interoperable programs and purposely make available the information needed to achieve interoperability.[134] One commentator asserts that a person may examine these publicly available materials in order to discover many of the ideas underlying the program.[135] This commentator notes that

[129] PARLIAMENT DEBATES, *see supra* note 126, at 27-28. Representative Cooney argued that the majority is under a misconception because "[i]t is not a question of trying to protect the big players on the stage, because the people who need protection are the individuals who are writing software programmes through their own intellectual ingenuity. We are now exposing them to the danger of having their work pirated and copied by opening the door [to piracy]." *Id.* at 28.

[130] BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 8. BSA is a member of SAGE and held the same legal position on the Proposed Directive. Telephone interview with Ms. Lori Forte, BSA representative (Nov. 15, 1990) (discussing BSA's position on reverse engineering provision).

[131] BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 8.

[132] *Id.* at 9.

[133] *Id.* at 11; Michael S. Keplinger, Official of the U.S. Patent and Trademark Office, *Ideas, Expression and Copyright Protection for Computer Programs*, at 12-13 (Mar. 28, 1989) (paper), *quoted in* BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 11. Mr. Keplinger states that "a person may examine publicly available materials for a copyrighted computer program in order to discover many of the ideas and concepts underlying that program. Running test data and other operations to exercise the program's functions can disclose much other information about the makeup and functions implemented in the program." Keplinger, *supra*, at 12-13.

[134] Telephone interview with Marilee Mott, WordPerfect representative (Jan. 28, 1991) (discussing WordPerfect's 'Developer's Toolkit,' which aids third party authors in creating interoperable programs).

[135] Keplinger, *supra* note 133, at 12-13; *see* BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 11 (containing selected text of Keplinger paper which lists alternatives to reverse engineering for achieving interoperability).

in addition to examining published materials, a programmer can also run test data through the program and examine the results to understand how to achieve interoperability.[136]

III. *U.S. LAW ON REVERSE ENGINEERING*

Unlike the Directive, U.S. copyright law does not explicitly permit reverse engineering.[137] Rather, U.S. copyright law establishes by statute exclusive rights for all copyright holders and places limitations on those rights.[138] The statutory limitations on these exclusive rights arguably do not allow for reverse engineering.[139] The limited U.S. case law on reverse engineering computer programs does not definitively indicate whether U.S. copyright law permits reverse engineering.[140]

A. *U.S. Statutory Law on Reverse Engineering: A Comparison with the Directive*

Like the Directive, the U.S. Copyright Act grants copyright holders the exclusive right to reproduce and distribute software.[141] The U.S. Copyright Act also confers on the author the right to create and authorize derivative works.[142] This exclusive right allows copyright holders to make or authorize all translations and adaptations of their work.[143] The Directive, in comparison, also confers on the author the right to make or authorize translations and adaptations of the program but does not specifically refer to these acts as derivative

[136] Keplinger, *supra* note 133, at 12-13; *see* BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 11.

[137] *See* 17 U.S.C. § 106 (1988); *supra* note 7 (containing text of section 106). Section 106 grants exclusive rights to the copyright holder, none of which refers to reverse engineering. 17 U.S.C. § 106 (1988). Nor do the limitations on these rights refer to reverse engineering. *Id.* § 117; *see supra* note 7 (containing text of section 117).

[138] 17 U.S.C. § 117 (1988); *supra* note 7 (containing text of section 117).

[139] 17 U.S.C. § 117 (1988).

[140] *See infra* notes 203-37 (discussing unsettled case law on reverse engineering of computer programs).

[141] *See* 17 U.S.C. § 106 (1988); *supra* note 7 (containing text of section 106 which grants rights of reproduction and distribution); *see also* Directive, *supra* note 2, art. 4, at 44 (granting exclusive rights to copyright owner); *supra* note 79 (containing text of article 4).

[142] 17 U.S.C. § 106(2) (1988); *see id.* § 101 (defining derivative works).

[143] *Id.* § 101; *see supra* note 25 (containing text of section 101).

works.[144]

In 1980, the U.S. Congress amended the Copyright Act in order to grant exclusive rights to authors of computer programs.[145] The purpose of the amendments was to encourage the development of the software industry in the United States.[146] Like the Community, the U.S. Congress recognized the need to place limitations on the exclusive rights granted to the copyright holder.[147] The 1980 amendments thus established two narrow exceptions to the exclusive rights held by a copyright owner.[148] First, the 1980 amendments allow users to copy a program for archival purposes.[149] Second, the amended Copyright Act permits copying if it is an essential step in the utilization of the program.[150] The Directive, on the other hand, permits reproduction and translation of the object code if it is necessary to achieve interoperability.[151] Furthermore, the Directive permits copying through decompiling and reverse engineering.[152] The U.S. Copyright Act has no such provision.[153]

The U.S. Copyright Act contains an additional exception, known as the fair use doctrine.[154] The fair use doctrine ex-

[144] *See* Directive, *supra* note 2, art. 4, at 44 (containing no specific reference to derivative works, but confers the rights to make adaptations and translations to the author); *supra* note 79 (containing text of article 4).

[145] 17 U.S.C. §§ 101-17 (1988); *see* R. NIMMER, *supra* note 1, at 10-12.

[146] *See* CONTU REPORT, *supra* note 1, at 11. The Commission recognized a need "to encourage the creation and broad distribution of computer programs in a competitive market." *Id.*

[147] *Compare* 17 U.S.C. § 117 (1988) *with* Directive, *supra* note 2, art. 5, at 44-45 (placing restrictions on exclusive rights granted to copyright owners). The Community also has granted an exception for decompiling to create interoperable programs. Directive, *supra* note 2, art. 6, at 45; *see supra* note 11 (containing text of article 6).

[148] 17 U.S.C. § 117 (1988); *see supra* note 7 (containing text of section 117).

[149] 17 U.S.C. § 117(2) (1988).

[150] *Id.* § 117(1).

[151] Directive, *supra* note 2, art. 6, at 45; *see supra* note 11 (containing text of article 6).

[152] Directive, *supra* note 2, art. 6, at 45. The exception permits the decompiling of object code into source code, and then allows a third party to create a new interoperable program. *Id.* This act of decompiling and creating a new program is reverse engineering. *See supra* notes 53-70 and accompanying text (explaining act of reverse engineering).

[153] *See supra* note 137 (discussing absence of reverse engineering provision in U.S. copyright statute).

[154] *See* 17 U.S.C. § 107 (1988). Section 107 considers four factors in determining whether the use is a fair use:

empts such unauthorized acts as reproduction and adaptation that would otherwise infringe a copyright.[155] There is no such exception in the Directive.[156] Whether a particular use is fair or not is to be determined by analyzing four factors: the purpose of the use, the nature of the copyrighted work, the amount of copyrighted material used, and the effect of the use on the commercial market for the original.[157]

The fair use doctrine does not give weight to any one particular factor and thus several interpretations have arisen.[158] One commentator suggests that the primary issue in determining fair use is the effect on the potential market for the original work and whether the secondary work and the original are functionally similar.[159] The U.S. Supreme Court has held that, when determining fair use, harm is presumed if the secondary use is commercial.[160] The application of the fair use doctrine thus depends on the nature of the secondary use.[161] A commercial secondary use tends to weigh against a finding of fair use.[162] Commentators, however, assert that a commercial use should merely weigh against a fair use finding, and not pre-

> (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.

Id.; *see* R. NIMMER, *supra* note 1, at 67. The fair use doctrine was originated by the courts and has been recognized in section 107 of the Copyright Act. *Id.* Professor Nimmer states that "[f]air use redefines or exempts behavior such as reproduction, adaptation, and the like that would otherwise infringe an underlying copyright." *Id.*; *see* Harper & Row Publishers, Inc. v. Nation Enter., 471 U.S. 539, 566 (1985). According to the U.S. Supreme Court, market impact is "undoubtedly the single most important element of fair use." *Id.*

[155] *See* R. NIMMER, *supra* note 1, at 69.

[156] *See* Directive, *supra* note 2. There is no reference to a fair use provision in the directive. *See id.*

[157] *See* R. NIMMER, *supra* note 1, at 67; *see also* 17 U.S.C. § 107 (1988).

[158] *See* R. NIMMER, *supra* note 1, at 67; *see also Reverse Engineering*, *supra* note 8, at 139-41; U.S. Ambassador Note, *supra* note 12, at 4-6.

[159] *See* R. NIMMER, *supra* note 1, at 68.

[160] *See* Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 451 (1984). The U.S. Supreme Court stated that "[e]very commercial use is presumptively an unfair exploitation of the monopoly privilege that belongs to the owner of a copyright." *Id.*

[161] *See* R. NIMMER, *supra* note 1, at 69.

[162] *See* Harper & Row Publishers, Inc. v. Nation Enter., 471 U.S. 539, 562 (1985).

clude one.[163]

Both the Directive and the U.S. Copyright Act protect the expression and not the idea embodied in a copyrighted work.[164] The Directive, however, fails to categorize the object and source code as either idea or expression.[165] Similarly, the U.S. Copyright Act also fails to address this issue.[166] U.S. courts have found, however, that the object code, source code, and sequence, structure and organization (the "SSO") of a computer program are protected by copyright.[167] As long as there is more than one way to express the idea embodied in the code, U.S. courts protect the codes as expression.[168] If there is only one way to express an idea, U.S. courts hold that the expression merges with the idea and is no longer protected.[169]

The U.S. Copyright Act, unlike the Directive, has not taken a definitive stand on the act of reverse engineering.[170] The U.S. Copyright Act, however, does not permit unauthorized derivative works.[171] Moreover, unlike the Directive, the U.S. Copyright Act does not explicitly permit decompiling and

[163] *See Reverse Engineering, supra* note 8, at 139; U.S. Ambassador Note, *supra* note 12, at 5; *see also* Triangle Publications, Inc. v. Knight-Ridder Newspapers, Inc., 626 F.2d 1171, 1175 (5th Cir. 1980) (holding that commercial use was fair use).

[164] *See* Directive, *supra* note 2, art. 1(2), at 44; 17 U.S.C. § 102(b) (1988); *supra* note 26 (containing text of section 102(b)); *supra* notes 26-28 and accompanying text (discussing idea/expression dichotomy).

[165] *See* Directive, *supra* note 2, art. 1(2), at 44.

[166] *See* 17 U.S.C. § 102(b) (1988); *supra* note 26 (containing text of section 102(b)).

[167] *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1239 (3d Cir. 1986), *cert. denied,* 479 U.S. 1031 (1987); Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986).

[168] *See Whelan,* 797 F.2d at 1236; Morrisey v. Proctor & Gamble Co., 379 F.2d 675, 678-79 (1st Cir. 1967) (discussing merger doctrine in idea/expression dichotomy); CONTU REPORT, *supra* note 1, at 20. CONTU defined the idea/expression dichotomy in the software context to mean that "when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement." *Id.*

[169] *See Morrisey,* 379 F.2d at 678-79; Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986); *see also* CONTU REPORT, *supra* note 1, at 20. If there are a limited number of ways to express the principle, it becomes idea. *Id.*

[170] *See generally,* 17 U.S.C. §§ 101-17 (1988); Directive, *supra* note 2, art. 6 at 45.

[171] 17 U.S.C. § 106(2) (1988).

reverse engineering.[172] U.S. courts have only alluded to re-verse engineering in the computer software context, thus leaving the issue unresolved in the United States.[173]

B. *U.S. Case Law on Reverse Engineering*

Only a limited number of U.S. copyright cases specifically address reverse engineering in the computer software context.[174] Several cases that do not involve computers have permitted reverse engineering under trade secret law.[175] In computer copyright cases, courts have only tangentially raised the issue of reverse engineering, thus leaving the U.S. position on reverse engineering of computer programs unresolved.[176]

1. Reverse Engineering in Non-Software Cases

The U.S. Supreme Court has held that trade secret law permits reverse engineering.[177] In addition, U.S. courts have held that reverse engineering is permissible for works not protected by a federal patent.[178] The leading U.S. Supreme Court reverse engineering case involved the preemption of a state statute that prohibited reverse engineering for unpatented works.[179]

In *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*,[180] the Court held that a state statute may not grant patent-like protection in

[172] *See id.* § 117; *supra* notes 147-50 and accompanying text (discussing limitations on exclusive rights to computer authors).

[173] *See infra* notes 203-37 and accompanying text (discussing computer case law pertinent to reverse engineering).

[174] *See Reverse Engineering, supra* note 8, at 137; *supra* notes 203-37 and accompanying text (discussing limited computer case law on reverse engineering).

[175] *See* Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470 (1974); Chicago Lock Co. v. Fanberg, 676 F.2d 400 (9th Cir. 1982).

[176] *See* Hubco Data Prods. Corp. v. Management Assistance, Inc., 219 U.S.P.Q. 450 (D. Id. 1983); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1501-02 n.17 (D. Minn. 1985).

[177] *See Kewanee*, 416 U.S. 470. The *Kewanee* Court permitted reverse engineering under trade secret law because trade secret law provides far weaker protection than patent law. *Id.* at 489-90. The Supreme Court has reaffirmed *Kewanee*, holding that "[t]he public at large remain[s] free to discover and exploit the trade secret through reverse engineering of products in the public domain or by independent creation." Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 155 (1989).

[178] *See Bonito Boats*, 489 U.S. at 141; Sears, Roebuck & Co. v. Stiffel Co., 376 U.S. 225 (1964); Compco Corp. v. Day-Brite Lighting, Inc., 376 U.S. 234 (1964).

[179] *See Bonito Boats*, 489 U.S. at 141.

[180] 489 U.S. 141 (1989).

the absence of federal patent protection.[181] The plaintiff Bonito Boats, Inc. ("Bonito Boats") created a wood and fiberglass boat hull.[182] Bonito Boats never filed a federal patent application to protect the design aspects of the hull or the manufacturing process.[183] After Bonito Boats' model had been on the market for six years, the Florida legislature enacted a statute that prohibited the direct molding process of unpatented boat hulls.[184]

Bonito Boats filed an action in Florida state court, alleging that the defendant, Thunder Craft Boats, Inc. ("Thunder Craft"), reverse engineered Bonito Boats' hull in violation of the state statute.[185] The U.S. Supreme Court affirmed the Florida Supreme Court and held for Thunder Craft.[186] The Court found that under the direct molding process, information obtained from the original hull was used to develop a new hull.[187] The Court held that this act of direct molding constituted reverse engineering.[188] The Court also held that the Florida statute conflicted with federal patent law and was, therefore, invalid under the Supremacy Clause of the U.S. Constitution.[189]

The Court held that the right to prohibit reverse engineering is one of the rights granted to federal patent holders, and not one that can be conferred to unpatented works by a state statute.[190] Because patent protection grants the owner a temporary monopoly right, the Court stressed the importance of retaining free competition for works that do not merit patent

[181] *See id.* at 156. The Court in *Bonito Boats* held that "[s]tates may not offer patent-like protection to intellectual creations which would otherwise remain unprotected as a matter of federal law." *Id.*

[182] *See id.* at 144.

[183] *See id.*

[184] *See id.* at 144-45.

[185] *See id.* at 145.

[186] *See id.* at 168.

[187] *See id.* at 144.

[188] *See id.* at 160. In *Bonito Boats,* Justice O'Connor stated that "the Florida law prohibits the entire public from engaging in a form of reverse engineering of a product in the public domain." *Id.*

[189] *See id.* at 144; *see* U.S. CONST. art. VI.

[190] Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 160 (1989). The Court held that "[reverse engineering] is clearly one of the rights vested in the federal patent holder, but has never been a part of state protection under the law of unfair competition or trade secret." *Id.*

protection.[191] The Court also stated that a work that does not merit federal protection should be available to the public.[192] Justice O'Connor emphasized that reverse engineering of articles in the public domain often leads to significant advances in technology.[193] For this reason, the Court held that a state statute that prohibited reverse engineering for unpatented works would hinder the advancement of technology.[194]

In *Secure Services Technology, Inc. v. Time and Space Processing, Inc.*,[195] the U.S. District Court for the Eastern District of Virginia ruled that the reverse engineering of a facsimile machine was permitted under trade secret law.[196] The plaintiff, Secure Services Technology, Inc. ("Secure Services"), manufactured a facsimile machine that emitted a coded digital pulse.[197] Secure Services' facsimile machines required that the receiving machines decode this digital pulse.[198] The defendant, Time and Space Processing, Inc. ("Time and Space"), decompiled this digital pulse and created a facsimile machine of its own that could decode Secure Services' digital signal, thus achieving interoperability.[199]

In addition to the trade secret action, Secure Services claimed that Time and Space violated its copyright.[200] The district court denied the copyright claim because the digital pulse that the Secure Services machine emitted did not meet

[191] *Id.* at 152; *see* Singer Mfg. Co. v. June Mfg. Co., 163 U.S. 169, 185 (1896). The *Bonito Boats* Court held that its

> past decisions have made clear that state regulation of intellectual property must yield to the extent that it clashes with the balance struck by Congress in our patent laws. The tension between the desire to freely exploit the full potential of our inventive resources and the need to create an incentive to deploy those resources is constant. Where it is clear how the patent laws strike that balance in a particular circumstance, that is not a judgment the States may second-guess.

Bonito Boats, 489 U.S. at 152.

[192] *Bonito Boats*, 489 U.S. at 160-61; *see supra* note 181 (quoting *Bonito Boats* Court in context of states granting patent-like protection).

[193] *Bonito Boats*, 489 U.S. at 160. Justice O'Connor stated that "reverse engineering of chemical and mechanical articles in the public domain often lead to significant advances in technology." *Id.*

[194] *See id.* at 160-61.

[195] 722 F. Supp. 1354 (E.D. Va. 1989).

[196] *Id.* at 1361.

[197] *Id.* at 1358.

[198] *Id.*

[199] *Id.* at 1358-59.

[200] *Id.* at 1359.

the authorship or originality requirements of U.S. copyright law.[201] Since trade secret law permits decompiling and reverse engineering, and because copyright protection was not applicable, the court permitted the defendant to market the interoperable facsimile machine.[202]

2. Reverse Engineering in Computer Software Cases

U.S. courts have not definitively addressed reverse engineering in the context of copyright protection for computer programs.[203] There are, however, several cases that have protected source code, object code, and SSO as expression.[204]

Hubco Data Products v. Management Assistance Incorporated[205] is one of numerous cases that have protected object code.[206] In *Hubco*, the U.S. District Court for the District of Idaho ruled

[201] *Id.* at 1363 n.25.

[202] *Id.* at 1365.

[203] *See infra* notes 203-37 (discussing limited case law on reverse engineering in computer programs); *see also* Atari Games Corp. v. Nintendo of Am. Inc., Nos. 88-4805-FMS, 89-0027-FMS, 89-0824-FMS, slip op. (N.D. Cal. Apr. 11, 1991). In *Atari v. Nintendo*, Nintendo of America Inc. and Nintendo Co., Ltd. ("Nintendo") sued Atari Games Corp. and Tengen, Inc. ("Atari") for copyright infringement. *Id.* at 4. The court enjoined Atari from "manufacturing, converting, copying, making any derivative of, producing, promoting, marketing, distributing, offering for sale, or selling either Nintendo's 10NES program . . . or anything substantially similar to [the] program." *Id.* at 17. The court found that Atari copied elements of Nintendo's program by "deprocessing" the chips used in Nintendo's program and by fraudulently obtaining a copy of Nintendo's copyrighted program from the Copyright Office. *Id.* at 2-3. By combining the information obtained from the "peeled" chips with the code obtained from the Copyright Office, Atari was able to create a program substantially similar to Nintendo's. *Id.* at 4-5. It appears that by enjoining Atari, the court was taking a stand against reverse engineering for commercial use in computer programs. *Id.* at 13-14. The court, however, explicitly rejects this inference in a footnote in the conclusion of the opinion. *Id.* at 17 n.2. The court stated that "[t]he prohibition on copying does not apply to reverse engineering by counsel and independent experts for purposes of this litigation." *Id.* This disclaimer is an example of the uncertainty in the U.S. courts on the issue of reverse engineering in copyrighted computer programs.

[204] *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1239 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1246-47 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 857 (2d. Cir. 1982); Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1498 n.11 (D. Minn. 1985).

[205] 219 U.S.P.Q. 450 (D. Id. 1983).

[206] *See, e.g. Whelan*, 797 F.2d at 1239; *Apple Computer*, 714 F.2d at 1246-47; *E.F. Johnson*, 623 F. Supp. at 1133.

that the object code in Management Assistance Incorporated's ("MAI") operating system program was subject to copyright protection.[207] The court further concluded that the idea/expression dichotomy did not preclude the code from copyright protection.[208]

In *Hubco*, MAI obtained a preliminary injunction against Hubco Data Products ("Hubco") to restrain Hubco from using a procedure that upgraded MAI's computer operating system.[209] The court found that Hubco's "upgrading" process involved two types of copyright infringement.[210] First, in developing the "upgrading" system, Hubco created a written printout of MAI's object code.[211] The court found this to be a clear violation of MAI's exclusive right to make reproductions of its program under the Copyright Act.[212] Second, Hubco developed a procedure that copied MAI's object code within the computer.[213] The court held that this action also violated MAI's exclusive rights.[214] The court found that Hubco's actions did not fall within the two exceptions set out in the Copyright Act.[215]

[207] *Hubco*, 219 U.S.P.Q. at 454. In granting a preliminary injunction the *Hubco* court found "that MAI has a reasonable probability of success on the issue of whether its object code is copyrightable under 17 U.S.C. § 102." *Id.*

[208] *Id.*

[209] *Id.* at 451. In *Hubco*, Hubco Data Products ("Hubco") brought suit alleging that MAI violated the antitrust laws. Hubco Data Prods. Corp. v. Management Assistance Inc., No. 81-1295 (D. Id. Aug. 2, 1984) (LEXIS, Genfed library, Dist file). MAI counterclaimed, asserting that Hubco had infringed MAI's copyright. *Id.* MAI sought a temporary restraining order and a preliminary injunction against Hubco. Hubco Data Prods. Corp. v. Management Assistance Inc., 219 U.S.P.Q. 450, 451 (D. Id. 1984). The court denied Hubco's antitrust motion and held in favor of MAI, granting a preliminary injunction against Hubco. *Id.* at 457.

[210] *Hubco*, 219 U.S.P.Q. at 456.

[211] *Id.* The court found that Hubco made

> a written printout of the unscrambled object code . . . . That written printout is certainly a 'material object' in which the copyrighted object code is 'fixed' and from which this higher-level object code can be 'reproduced or otherwise communicated either directly or with the aid of a machine.' Hubco uses that written printout to 'communicate' or 'reproduce' the higher-level object code.

Id.

[212] *Id.*

[213] *Id.* The court found that the "copying and comparison procedures . . . [were] done completely inside the computer by the software program. . . . The Court finds that it does not make a difference." *Id.*

[214] *Id.*

[215] *Id.* The court found that Hubco could not "avail itself" of the owner ex-

The court enjoined Hubco from marketing the program derived from MAI's software.[216] The court's holding was predicated upon the principle that object code is protected by copyright.[217] The court, in dicta, also reasoned that if Hubco had deciphered the code through reverse engineering it would have constituted an infringement of MAI's protected software.[218] The dicta in *Hubco* is the closest a court has come to addressing the issue of reverse engineering in copyright protected computer software cases.

In addition to protecting object code, U.S. courts hold that copying the source code and the SSO is an infringement of copyright law.[219] If the codes are protected by copyright law, a reasonable inference can be made that unauthorized decompiling of the object code would constitute an infringement.[220] The extent to which copyright law protects the codes and SSO against reverse engineering, however, remains unclear.[221] Cases that are referred to as anti-reverse engineering

emption discussed in section 117 of the copyright statute. *Id.*; 17 U.S.C. § 117 (1988).

[216] Hubco Data Prods. Corp. v. Management Assistance Inc., 219 U.S.P.Q. 450, 457-58 (D. Id. 1983).

[217] *Id.* at 454.

[218] *Id.* at 452. The court would have enjoined Hubco even if there were "a reasonable probability of successfully showing that it deciphered the code through reverse engineering rather than through misappropriation." *Id.*

[219] *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1239 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1246-47 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 857 (2d. Cir. 1982); Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1498 n.11 (D. Minn. 1985).

[220] *See* U.S. Ambassador Note, *supra* note 12, at 1-2; *see also* Reback & Hayes, *U.S. Companies Join European Software Debate*, COMPUTER LAW STRATEGIST, Feb. 1990, at 5.

[221] *See* U.S. Ambassador Note, *supra* note 12, at 4. Since the issue of decompiling has not been specifically addressed by the courts, there is no case history available on how a court would consider a defendant's need to gain access to the idea of a computer program, by decompiling the code. *Id.* *But see* Manufacturers Technologies, Inc. v. Cams, Inc., 706 F. Supp. 984 (D. Conn. 1989). *Manufacturers* involved an action for a copyright infringement of a computer screen display. *Id.* at 990. The court, in dicta, recommended that in order for a plaintiff to prove copyright infringement it would have to show substantial similarity in the screen display as well as the program codes. *Id.* at 993. The court stated that "[t]he downfall of such an approach is that if, in fact, the infringing party only reverse engineered the screen displays themselves without having had access to the source or object codes, then the plaintiff's task in showing infringement of the computer program itself, would be difficult,

do not take a definitive stand on the issue.[222] Similarly, cases cited as favoring reverse engineering allude to the act of reverse engineering only in dicta.[223]

In *E.F. Johnson Co. v. Uniden Corporation of America*,[224] the U.S. District Court for the District of Minnesota held that Uniden Corporation of America ("Uniden") infringed E.F. Johnson Co.'s ("E.F. Johnson") copyrighted software when it disassembled the program and developed a substantially similar product.[225] The court held in favor of E.F. Johnson, noting the strong evidence of direct copying.[226] The court indicated in dictum, however, that reverse engineering does not in itself constitute misappropriation.[227]

In *Vault Corporation v. Quaid Software Limited*,[228] Vault Corporation ("Vault"), a producer of an anticopying computer program, brought a copyright infringement action against Quaid Software Limited ("Quaid"), a designer of a program that "unlocked" the protective lock created by Vault.[229] In developing the program that would "unlock" Vault's program, Quaid had to copy part of Vault's program.[230] The U.S. Court of Appeals for the Fifth Circuit found that Quaid's copying did not infringe Vault's copyright.[231] The court reasoned that Quaid's copying was protected by the Copyright Act.[232] The

if not insurmountable." *Id.* An inference can be made from this statement that the decompiling of the object code is prohibited, because in order to prove an infringement, defendant had to actually copy the code. *Id.*

[222] *See Hubco*, 219 U.S.P.Q. 450 (D. Id. 1983); *Manufacturers*, 706 F. Supp. 984; Atari Games Corp. v. Nintendo of America Inc., Nos. 88-4805-FMS, 89-0027-FMS, 89-0824-FMS, slip op. (N.D. Cal. Apr. 11, 1991).

[223] *See E.F. Johnson*, 623 F. Supp. at 1501-02 n.17; *see also* Vault Corp. v. Quaid Software Ltd., 847 F.2d 255 (5th Cir. 1988). Commentators cite *E.F. Johnson* and *Vault* as pro-reverse engineering cases. *See* D.C. Toedt III, *supra* note 64, at 2:19-21; Reback & Hayes, *supra* note 220, at 4.

[224] 623 F. Supp. 1485 (D. Minn. 1985).

[225] *Id.* at 1492.

[226] *Id.* at 1493-98.

[227] *Id.* at 1501-02 n.17. The court in dictum stated that "[t]he mere fact that defendant's engineers dumped, flow charted, and analyzed plaintiff's code does not, in and of itself, establish pirating. As both parties' witnesses admitted, dumping and analyzing competitors' codes is a standard practice in the industry." *Id.*

[228] 847 F.2d 255 (5th Cir. 1988).

[229] *Id.* at 256-58.

[230] *Id.* at 257.

[231] *Id.* at 270.

[232] *Id.* at 261. The court in *Vault* held that "the copy made by Quaid *was* 'created as an essential step in the utilization' of Vault's program. Section 117(1) con-

Copyright Act, stated the court, immunizes from liability those copies of computer programs that are made as an essential step in the utilization of the program.[233] Because Quaid only copied Vault's program as a step in using the program, the court did not find an infringement.[234]

Commentators cite *Vault* as a case that permits reverse engineering.[235] Like *Bonito Boats*, however, *Vault* concerned a state statute that prohibited reverse engineering.[236] The *Vault* court took the view that a state law outlawing reverse engineering touches upon an area exclusively controlled by federal law, and is therefore preempted.[237]

Despite the detailed statutory law in the computer software context, present U.S. case law fails to take a definitive stand on whether reverse engineering falls within the scope of copyright protection.

IV. *REVERSE ENGINEERING SHOULD BE PROHIBITED*

A. *U.S. Law Prohibits Reverse Engineering*

The U.S. Copyright Act arguably prohibits reverse engineering in computer software cases.[238] The U.S. Copyright Act explicitly prohibits the creation of unauthorized derivative works.[239] Even if reverse engineering resulted in the creation of an interoperable program, the interoperable work would probably constitute an unauthorized derivative work.[240] An unauthorized interoperable program would violate the U.S. copyright statute's proscription against unauthorized deriva-

tains no language to suggest that the copy it permits must be employed for a use intended by the copyright owner, and, absent clear congressional guidance to the contrary, we refuse to read such limiting language into this exception." *Id.* (emphasis in original).

[233] *See id.*; *see also* 17 U.S.C. § 117 (1988); *supra* note 7 (containing text of section 117).

[234] Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 261 (5th Cir. 1988).

[235] *See* D.C. TOEDT III, *supra* note 64, at 2:20-21; *see also* Reback & Hayes, *supra* note 220, at 4.

[236] *See Vault*, 847 F.2d at 269; *see also* Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 168 (1989).

[237] *Vault*, 847 F.2d at 269.

[238] *See supra* notes 141-69 (discussing U.S. statutory law in the context of reverse engineering).

[239] *See* 17 U.S.C. § 106(2) (1988); *supra* note 7 (containing text of section 106(2)); *supra* notes 142-43 and accompanying text (discussing derivative works).

[240] *See* U.S. Ambassador Note, *supra* note 12, at 2.

tive works.[241]

The Copyright Act permits unauthorized copying for archival purposes, or as an essential step in the utilization of the program.[242] Reverse engineering does not fall into either of these two exceptions.[243] Because reverse engineering is not mentioned in the Copyright Act, creators are therefore able to interpret the silence of the statute in favor of their position.[244] Some commentators continue to advocate reverse engineering despite the statements of the U.S. Ambassador to the Community, who has asserted that reverse engineering constitutes an unauthorized reproduction or translation of substantial parts of a program's object code.[245] According to the U.S. Ambassador to the Community, such a clear unauthorized reproduction represents a prima facie violation of the owner's right to reproduce and make derivative works.[246]

U.S. law employs the fair use doctrine to balance the need to protect an author's investment against the public's interest in the dissemination of information.[247] Whether the reverse engineering of object code constitutes a fair use depends upon the analysis of the four statutory fair use factors.[248] One commentator argues that a fair use defense should not be available where the reverse engineer copies an entire program, for financial gain, in order to produce a competitive software product.[249] Conversely, a fair use should be found when a computer science student reverse engineers a program simply to learn programming techniques.[250] Somewhere between these two extremes is reverse engineering that involves copying an entire program, for financial gain, but only in order to achieve

[241] *Id.*

[242] *See* 17 U.S.C. § 117 (1988); *supra* note 7 (containing text of section 117).

[243] *See* 17 U.S.C. § 117 (1988); U.S. Ambassador Note, *supra* note 12, at 3.

[244] *See* 17 U.S.C. § 117 (1988); *supra* note 7 (containing text of section 117). The defendant in *Vault*, Quaid Software Ltd., "contends that its activities fall within the § 117 exceptions and that it has, therefore, not infringed." Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 259 (5th Cir. 1988).

[245] U.S Ambassador Note, *supra* note 12, at 2; *see* D.C. Toedt III, *supra* note 64, at 2:19; Reback & Hayes, *supra* note 220, at 4.

[246] U.S. Ambassador Note, *supra* note 12, at 2.

[247] *See* 17 U.S.C. § 107 (1988); *supra* note 154 (containing text of section 107); *supra* notes 154-57 and accompanying text (discussing fair use doctrine).

[248] 17 U.S.C. § 107 (1988).

[249] *See* Grogan, *supra* note 12, at 10.

[250] *Id.*

interoperability.[251] If the underlying purpose of the reverse engineering is to legitimately copy ideas rather than expression, a court might be more willing to accept a fair use defense.[252]

The U.S. Supreme Court has held that a commercial use does not weigh in favor of finding a fair use.[253] The language of the Supreme Court does not, however, mandate that interoperable competitive programs cannot be a fair use.[254] Considering the need for factual application in the statute, the United States arguably has not taken a definitive position on whether reverse engineering to achieve interoperability is a fair use.[255] The Directive goes beyond the scope of the fair use doctrine and permits a reverse engineer to decompile to create interoperable competitive programs.[256]

Since the U.S. Copyright Act only protects the expression, the public is not prohibited from accessing the ideas.[257] Like the Directive, the statute fails to categorize computer code as idea or expression.[258] U.S. courts, however, interpret the copyright statute to protect object code, source code and SSO as expression.[259] Cases that protect the codes and the SSO involve bad faith misappropriation for commercial use.[260] Simi-

[251] *Id.*

[252] *See id.*

[253] *See* Harper & Row Publishers, Inc. v. Nation Enter., 471 U.S. 539, 562 (1985); Sony Corp. of America v. Universal City Studios, Inc., 464 U.S. 417, 449 (1984).

[254] *See* U.S. Ambassador Note, *supra* note 12, at 5. Ambassador Niles stated that "[t]he legislative history of the 1976 Act affirms that the privilege of fair use is potentially applicable to all categories of users, notwithstanding the apparently preferred status accorded [to] non-commercial educational and scholarly uses." *Id.*

[255] *See generally, id.*

[256] Directive, *supra* note 2, art. 6, at 45; *see supra* note 11 (containing text of article 6).

[257] 17 U.S.C. § 102(b) (1988); *see supra* note 26 (containing text of section 102(b)).

[258] *See supra* notes 158-63 and accompanying text (discussing idea/expression dichotomy and its application in U.S. courts).

[259] *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1239 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1246-47 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 857 (2d. Cir. 1982); Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1498 n.11 (D. Minn. 1985).

[260] *See, e.g.*, *Whelan*, 797 F.2d 1226; *Apple Computer*, 714 F.2d 1240; *Stern Elecs.*,

larly, U.S. courts may prevent reverse engineering of the codes in cases that involve a bad faith misappropriation.[261]

The court in *Hubco* protected the program's object code as well as stating in dicta that even if the code was deciphered through reverse engineering, the code would still be protected.[262] Thus, *Hubco* stands for the proposition that copyright law does not permit reverse engineering.

Despite the clarity of the statute and the judicial protection of object code, source code, and SSO, some commentators argue that reverse engineering is permitted under U.S. law.[263] These commentators cite *Bonito Boats* as a case permitting reverse engineering.[264] Analysis of the case, however, shows that *Bonito Boats* does not approve of reverse engineering, but rather emphasizes the importance of obtaining federal intellectual property protection.[265] By invalidating the state statute that prohibited reverse engineering, the Court in *Bonito Boats* held that state statutes cannot supplant federal power to grant intellectual property protection.[266]

Similarly, commentators rely on *Vault* as a case that permits reverse engineering.[267] This case also involved a state statute that prohibited reverse engineering.[268] In invalidating the state statute, the court did not infer that U.S. law permits reverse engineering.[269] In *Vault*, the court permitted Quaid to copy Vault's program only because the act of copying was a prerequisite for using the program, and was thus within the scope of the Copyright Act.[270] The court did not address

669 F.2d at 853; Hubco Data Prods. Corp. v. Management Assistance Inc., 219 U.S.P.Q. 450 (D. Id. 1983).

[261] *See* U.S. Ambassador Note, *supra* note 12, at 5-6.

[262] *Hubco*, 219 U.S.P.Q. at 452; *see supra* notes 205-18 (discussing *Hubco*).

[263] *See* D.C. TOEDT III, *supra* note 64, at 2:59-62; Levine, *Comment on Follow-Up*, 6 COMPUTER LAW. 29 (July 1989); Reback & Hayes, *supra* note 220, at 4.

[264] D.C. TOEDT III, *supra* note 64, at 2:31; Levine, *supra* note 263, at 29; Reback & Hayes, *supra* note 220, at 4. Toedt concludes that "[t]he *Bonito Boats* opinion praised reverse engineering for its innovative character." D.C. TOEDT III, *supra* note 64, at 2:31.

[265] Perle, Meyer & Siber, *Bonito Boats Redux*, 7 COMPUTER LAW. 2-3 (Feb. 1990).

[266] *Id.* at 2-3.

[267] *See* D.C. TOEDT III, *supra* note 64, at 2:20-21; Reback & Hayes, *supra* note 220, at 4.

[268] Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 268-70 (5th Cir. 1988).

[269] *See id.*; *Reverse Engineering*, *supra* note 8, at 148-49.

[270] *Vault*, 847 F.2d at 758.

whether it would permit copying for reverse engineering purposes.[271] The court's silence on reverse engineering thus does not connote its approval of the act.[272]

In *Secure Services*, the court permitted decompiling and reverse engineering to create an interoperable facsimile machine under trade secret law.[273] This court, however, did not address whether decompiling and reverse engineering were permitted under copyright law.[274] Secure Services' digital pulse did not have the requisite level of originality needed for copyright protection.[275] The court thus did not analyze reverse engineering in the context of copyright law.[276]

The case law on reverse engineering in computer programs is limited.[277] Proponents of reverse engineering must therefore rely upon dicta to support their position.[278] The *E.F. Johnson* decision has been cited as favoring reverse engineering.[279] This reliance, however, is misplaced. The court in *E.F. Johnson* held the defendant liable for disassembling the plaintiff's program and developing a substantially similar program.[280] The facts and holding clearly prohibit reverse engineering, and only vague dictum states reverse engineering is not a form of misappropriation.[281]

B. *The Directive Should Not Have Permitted Reverse Engineering*

The Directive explicitly permits decompiling to achieve interoperability.[282] The goal of creating interoperable programs is desirable, but a detailed provision that permits decompiling

[271] *See id.*

[272] *See id.*

[273] Secure Servs. Technology, Inc. v. Time and Space Processing, Inc., 722 F. Supp. 1354, 1365 (E.D. Va. 1989).

[274] *Id.* at 1365.

[275] *Id.* at 1363.

[276] *Id.*

[277] *See* U.S. Ambassador Note, *supra* note 12, at 4.

[278] *See* D.C. TOEDT III, *supra* note 64, at 2:19; Reback & Hayes, *supra* note 220, at 4.

[279] *See* D.C. TOEDT III, *supra* note 64, at 2:19; Reback & Hayes, *supra* note 220, at 4.

[280] E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485 (D. Minn. 1985).

[281] *Id.* at 1501-02 n.17; *see supra* note 227 (containing dicta from *E.F. Johnson*).

[282] Directive, *supra* note 2, art. 6, at 45; *see supra* note 11 (containing text of article 6).

is not the best way of guaranteeing interoperability.[283]

Small software authors were concerned that if interoperability were not specifically provided for in the Directive, larger software manufacturers would form a monopoly against them.[284] As the Commission stated in the Proposed Directive, Community competition law prohibits dominant manufacturers from forming monopolies.[285] A specific exception for decompiling to achieve interoperability therefore becomes unnecessary if Community competition law is interpreted together with the Directive's copyright protection.[286]

The fear that larger software manufacturers will monopolize their strong market position is unwarranted.[287] Prominent software manufacturers already publicly distribute the information needed to create interoperable programs.[288] Additionally, the third party author has numerous alternatives to decompiling that provide the information needed to create an interoperable program.[289]

Other software manufacturers fear that the inclusion of such a provision will unlock the door to piracy.[290] The Community, however, should follow the course taken by the United States. The United States has a broad copyright statute applicable to computer software that does not specifically provide for decompiling.[291] Instead of granting an exception as does the Directive, the United States grants the author the exclusive right to create derivative works.[292] This exclusive right does not permit the unauthorized creation of interoperable pro-

[283] *See supra* notes 119-36 and accompanying text (discussing arguments against reverse engineering).

[284] *See supra* notes 106-08 and accompanying text (discussing fear of monopoly if reverse engineering is prohibited).

[285] *See supra* note 93 and accompanying text (discussing EC competition law as alternative to reverse engineering provision).

[286] *See* BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 11-12.

[287] *Id.*

[288] *See supra* notes 133-36 and accompanying text (discussing alternatives to reverse engineering).

[289] *Id.*

[290] *Defense Against Pirates*, *supra* note 62, at 138.

[291] 17 U.S.C. §§ 101-117 (1988); U.S. Ambassador Note, *supra* note 12, at 1; *see also* BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 3.

[292] 17 U.S.C. § 106(2) (1988); *see supra* note 7 (containing text of section 106(2)).

grams.[293]

The U.S. Copyright Act, however, has a fair use doctrine that permits third parties to infringe upon this exclusive right if the court determines that the use complies with the statute's four factors.[294] A narrow interpretation of the fair use doctrine would classify an interoperable program as a permitted commercial use.[295] If the program is truly interoperable and not substantially similar, rather than hinder the market value of the original work it will enhance the original work's value.[296]

Promoting the European software industry and harmonizing the laws of the Member States are admirable goals, but the Directive falls short of achieving these goals in two areas.[297] First, the Directive does not define interoperability, an omission which could lead to confusion among software producers as to what does and does not constitute a legal interoperable program.[298] Moreover, in codifying the idea/expression dichotomy, the Directive does not explicitly dictate whether the code is idea or protected expression.[299] Although the U.S. Copyright Act is also silent with respect to this classification, U.S. courts have unanimously protected code as expression.[300] The Community might not be so fortunate, thus leaving open the opportunity for potential conflict among the Member States.

The dangers of permitting reverse engineering are made obvious by the safety provisions included in article 6.[301] The

[293] *See supra* notes 239-41 and accompanying text (discussing interoperable programs as unauthorized derivative works).

[294] *See* 17 U.S.C. § 107 (1988); *supra* 247-52 and accompanying text (discussing prohibition of reverse engineering under fair use doctrine).

[295] R. NIMMER, *supra* note 1, at 67-68; *Reverse Engineering*, *supra* note 8, at 138-41.

[296] U.S. Ambassador Note, *supra* note 12, at 6.

[297] *See infra* notes 298-99 and accompanying text (discussing shortcomings of Proposed Directive).

[298] *See supra* notes 119-25 and accompanying text (discussing SAGE's argument about vagueness of term interoperability).

[299] Directive, *supra* note 2, art. 1, at 44. A presumption can be made that the object and source codes are protected expressions because if they were not protected, there would be little need for a decompilation provision.

[300] 17 U.S.C. § 102(b) (1988); *supra* note 26 (containing text of section 102(b)); *supra* notes 167-69 and accompanying text (discussing U.S. court's classification of codes and SSO as protected expression).

[301] Directive, *supra* note 2, art. 6, at 45; *see supra* note 11 (containing text of article 6).

Community has gone to great lengths to attempt to protect software authors from the misuse of reverse engineering.[302] Broad copyright protection prohibiting reverse engineering would be more effective than the present Directive's combination of both allowing and limiting reverse engineering. Moreover, the Community should rely upon its competition law to guard against monopolistic situations.[303]

CONCLUSION

The United States has successfully protected its authors from piracy. Having been protected by copyright law, U.S. software makers readily offer the necessary information needed to create interoperable programs. U.S. software authors continue to develop new programs on the premise that copyright law protects them from being decompiled by pirates.

The goal of the Community's Directive is to protect computer software so that the European computer industry will flourish. It appears that the United States has accomplished this goal. The Community would do well to take note of the U.S. copyright law and recognize that its inclusion of a specific provision permitting reverse engineering was misguided. The present Directive is not necessary to ensure the creation of interoperable programs, and may very well lead to the increased piracy of computer software.

Mindy J. Weichselbaum[*]

[302] *See supra* notes 91-101 and accompanying text (discussing history of reverse engineering provision and attempts EC has taken to insure its proper application).

[303] Commission's Proposed Directive, *supra* note 3, O.J. C 91/4, at 16; BSA REVERSE ENGINEERING PAPER, *supra* note 104, at 12.

[*] J.D. Candidate, 1992, Fordham University.