[Vanderbilt Journal of Transnational Law](https://scholarship.law.vanderbilt.edu/vjtl)

Volume 25
Issue 2 *Issue 2 - 1992*

Article 7

5-1992

The EC Directive on the Legal Protection of Computer Programs

Linda G. Morrison

The EC Directive on the Legal Protection of Computer Programs: Does It Leave Room for Reverse Engineering Beyond the Need for Interoperability?

ABSTRACT

The evolution of computer technology has launched questions regarding the proper scope of protection for computer software. The European Community (EC) recently adopted a Council Directive on the Legal Protection of Computer Programs (the Directive), which protects computer software under the copyright paradigm. The path to final adoption of the Directive, however, was marked by debates between diametrically opposed lobbying groups regarding the propriety of a reverse engineering exception to the exclusive right of reproduction. This Note discusses the lobbying efforts that led to a compromise and analyzes the Directive through a comparison to United States law. Next, the Note analyzes a "look and feel" infringement suit under the Directive in an attempt to discover current trends in the international copyright protection of computer software. The Note concludes that the Directive is a laudable step toward legitimizing the process of reverse engineering and promoting international standards of protection. Nonetheless, the author concludes that the decompilation exception found in the Directive is overly limited by the requirement that decompilation be indispensible to interoperability. The author argues that a broader reverse engineering right to discover underlying ideas would have better promoted the EC computer industry's desire to break into the international software market.

TABLE OF CONTENTS

I. INTRODUCTION 294
II. ADOPTION OF A COMPROMISE DIRECTIVE.......... 298
A. *Stated Purpose of the Directive* 298
B. *Precompromise Lobbying* 300
C. *The Resulting Compromise Directive* 305
1. Protectible Subject Matter and Exclusive Rights.............................. 307
2. Exceptions to Exclusive Rights 310

a. Article 5—Observe, Study, and Test Exception 310

b. Article 6—Explicit Decompilation Exception 315

III. STATUS OF UNITED STATES LEGAL PROTECTION OF COMPUTER PROGRAMS 318

A. *Evolving Scope of Protection for Computer Software* 319

B. *The Reverse Engineering Debate Under the Copyright Act* 321

C. *United States Concern for Interoperability* 324

IV. ANALYSIS OF A "LOOK AND FEEL" INFRINGEMENT SUIT UNDER THE DIRECTIVE 326

A. Lotus Development Corp. v. Paperback Software Int'l 327

B. *The "Look and Feel" Debate* 328

C. *The Facts of* Lotus *Analyzed under the Directive* 329

V. CONCLUSION 331

I. INTRODUCTION

The development and evolution of computer technology has yielded a highly competitive global industry for computer hardware and software.[1] Such a competitive environment has launched questions regarding the proper scope of protection that should be granted to computer programs.[2] The United States has chosen copyright as the proper mode of protection.[3] Similarly, the European Community (EC or the Community),[4] recognizing the impact that the software industry has on international trade, has made the protection of computer programs part of the 1992

[1] DIRK SCHROEDER, COMPUTER SOFTWARE PROTECTION AND SEMICONDUCTOR CHIPS at preface (1990).

[2] *See infra* part, III.A.

[3] *See* 17 U.S.C. § 101 (1988) (definition of computer program). Under United States law, computer programs are given full copyright protection as literary works. *See id.* §§ 101, 102. This hardline copyright approach of the United States has been described critically as "maxiprotectionist." J.H. Reichman, *Computer Programs as Applied Scientific Know-How: Implications of Copyright Protection for Commercialized University Research*, 42 VAND. L. REV. 639, 699 n.312 (1989).

[4] The European Community includes 12 member states: Belgium, Denmark, France, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal, Spain, the United Kingdom, and Germany. *See generally* 1 LAW OF THE EUROPEAN COMMUNITIES para. 1.01 to 1.18 (discussing initial formation of and subsequent accessions to the EC).

unification effort.[5]

The EC Commission[6] began its effort by analyzing the economic and technological impact of computer software on the Community.[7] The EC Commission, in its Green Paper on Copyright and the Challenge of Technology (Green Paper), concluded that computer programs should be protected under the copyright paradigm.[8] This initial Green Paper on computer software ultimately led to the adoption[9] on May 14, 1991 of a Council Directive on the Legal Protection of Computer Programs (Directive or Final Directive).[10]

[5] *See* Commission Proposal for a Council Directive on the Legal Protection of Computer Programs, COM(88)816 final, 1989 O.J. (C 91) 4 [hereinafter Proposed Directive]; *see also* SCHROEDER, *supra* note 1, at preface.

[6] The Commission, which proposes, drafts, and administers legislation, is a permanent body of civil servants that owes no allegiance to a single member state. *See* Treaty Establishing the European Economic Community, Mar. 25, 1957, arts. 155, 157, 189, 298 U.N.T.S. 11, 71-72, 78-79.

[7] *See* Commission of the European Communities, Green Paper on Copyright and the Challenge of Technology—Copyright Issues Requiring Immediate Action, COM(88)172 final at 170 (June 7, 1988).

[8] *Id.* at 179; *see also* Reichman, *supra* note 3, at 699 n.312.

[9] The path to adopting a directive is a complex one, involving several steps: (1) the Commission proposes a draft directive; (2) Parliament votes to adopt the directive; (3) the directive goes to the EC Council of Ministers for an opinion; (4) the proposal goes back to Parliament for a second reading; and (5) formal adoption by the EC Council—the final hurdle. *See EC Parliament Adopts Software Directive with Reverse-Engineering Compromise*, 40 Pat. Trademark & Copyright J. (BNA) 293, 293 (Aug. 2, 1990). Because the road to final adoption is so lengthy, lobbying groups have ample opportunity to influence the final version of the directive. *See infra* part II.B; *see also infra* note 10.

[10] Council Directive on the Legal Protection of Computer Programs, 1991 O.J. (L 122) 42 [hereinafter Final Directive].

The timetable for the adoption of the Council Directive on the Legal Protection of Computer Programs (Directive or Final Directive) went as follows: On January 5, 1989, the Commission submitted a proposal for a directive along with an explanatory memorandum. *See* Proposed Directive, *supra* note 5, 1989 O.J. (C 91) at 4-16. On October 18, 1989, the Economic and Social Committee, at the request of the Council of Ministers, adopted an opinion on the proposed Directive. *See* 1989 O.J. (C 329) 4. On July 11, 1990, the European Parliament, on its first reading of the proposed Directive, submitted amendments including, for the first time, a new article 5a, which allowed reverse engineering when necessary to ensure the creation of interoperable programs. *See* 1990 O.J. (C 231) 78, 81 [hereinafter Parliament Amendments]. On October 18, 1990, the Commission submitted an amended proposal incorporating Parliament's suggestions to the Council of Ministers. *See* Amended Proposal for a Council Directive on the Legal Protection of Computer Programs, 1990 O.J. (C 320) 22 [hereinafter Amended Proposal]. On December 13, 1990, the EC Council of Ministers approved the amended proposal. *See EC Council of Ministers Approves Amended Software Directive*, COMPUTER LAW., Feb. 1991, at 36, 36. On its second reading, the Parliament made no further

The path to the final adoption of this Directive was a rocky one, involving disputes between various factions in the software industry regarding to what extent the exclusive rights of software copyright owners would be limited by exceptions.[11] The debates focused primarily on the propriety of granting a reverse engineering[12] exception to the exclusive right of reproduction. When a software engineer engages in reverse engineering, technically the engineer is making reproductions of the program being analyzed.[13] Thus, unless an explicit exception is recognized, all reverse engineering techniques would constitute acts of infringement. Because reverse engineering has become a common practice in the industry,[14] the debate over the right of competitors to reverse engineer an original product is not unique to the EC.

In the United States, leading commentators differ over whether the Copyright Act of 1976 (the Copyright Act),[15] and the case law developed thereunder, can be interpreted as permitting noninfringing reverse engineering.[16] Furthermore, the debate becomes complicated by recent United States decisions[17] protecting not only literal aspects of computer programs, but also nonliteral aspects such as the structure, sequence, and

amendments. *See* Thomas C. Vinje, *The Development of Interoperable Products Under the EC Software Directive*, COMPUTER LAW., Nov. 1991, at 1, 11 n.43. Finally, following Parliament's second reading, on May 14, 1991 the Council of Ministers adopted the Directive, which contains an express decompilation provision. *See* Final Directive, *supra*, 1991 O.J. (L 122) at 42-46; *see also EC Adopts Software Directive*, COMPUTER LAW., June 1991, at 39, 39.

[11] *See infra* part II.B.

[12] Reverse engineering involves observing the features of the original software product to determine what makes it competitive on the market. *See* Chris Reed, *Reverse Engineering Computer Programs Without Infringing Copyright*, 13 EUR. INTELL. PROP. DEV. 47, 47 (1991). For the purposes of this Note, reverse engineering will be used as an overarching term of art describing a universe of analysis techniques, including decompilation.

[13] Vinje, *supra* note 10, at 3.

[14] *See* Clifford G. Miller, *The Proposal for an EC Council Directive on the Legal Protection of Computer Programs*, 12 EUR. INTELL. PROP. REV. 347, 349 (1990); *The Proposed EC Directive on the Legal Protection for Computer Programs: Position Statement*, 6 COMPUTER L. & PRAC. 97, 100 (1990) [hereinafter *ECIS Position Statement*]; *see also* Vinje, *supra* note 10, at 3.

[15] Copyright Act of 1976, Pub. L. No. 94-553, 90 Stat. 2541 (1976) (codified as amended at 17 U.S.C. § 101 (1988)).

[16] *See infra* part III.B (discussing debate in the United States over the legality of reverse engineering).

[17] *See, e.g.*, Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990); Whelan Assocs. v. Jaslow Dental Lab., 797 F.2d 1222 (3d Cir. 1986).

organization of computer programs.[18] Copyright protection in general does not extend to ideas.[19] Additionally, the doctrine of merger[20] further limits the scope of copyright protection. If the organization of a computer program as well as the "look and feel"[21] of user interfaces[22] are protected under copyright, the issue then becomes whether competitors should be able to reverse engineer the original software product in attempts to determine the underlying ideas and principles that "make [the original program] competitive and attractive in the market."[23]

Another important issue is whether these later entrants into the software market should be able to reverse engineer commercially success-

[18] *See, e.g.*, *Whelan*, 797 F.2d at 1233-42, 1248 (protecting the structure, sequence, and organization of computer programs); Johnson Controls, Inc. v. Phoenix Control Systems, Inc., 886 F.2d 1173, 1175 (9th Cir. 1989) (nonliteral components of computer software may be protected by copyright when they constitute expression rather than ideas); Broderbund Software, Inc. v. Unison World, 648 F. Supp. 1127, 1133 (N.D. Cal. 1986) (relying on *Whelan* to protect structure, sequence, and arrangement of print screens).

[19] 17 U.S.C. § 102(b) ("In 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." (emphasis added)).

[20] The doctrine of merger refers to instances in which the underlying idea is tied to expression. By protecting expression in this instance, the copyright paradigm would grant the author a monopoly on the idea. Hence, when merger of idea and expression occurs, copyright protection should be limited. *See* 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[B][3] (1990).

[21] The "look and feel" of a software package refers to the nonliteral elements of a program. *See Lotus*, 740 F. Supp. at 62-63 (discussing the "look and feel" concept); *see also* Pamela Samuelson & Robert J. Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits*, 30 JURIMETRICS J. 121, 126-27 (1989) (noting that many professionals in the software industry are unsure of what exactly is meant by the phrase "look and feel"). For purpose of discussion, the "look and feel" of a computer spreadsheet refers to the user's overall perception of the particular spreadsheet package, the appearance of the screen, and the means of executing commands (whether by mouse selection, by the successive appearance of menus, or simply by the selection of highlighted written commands at the top of the display screen). This listing, however, is by no means exclusive.

[22] User interface refers to the elements of a computer program through which the user is able to interact with, and provide data to, the underlying program code that is being executed. Aspects of user interfaces include command terms, interface functionality, screen sequence, screen layouts, and use of icons. *See* Samuelson & Glushko, *supra* note 21, at 132; *see also* Ronald L. Johnston & Allen R. Grogan, *Copyright Protection for Command Driven Interfaces*, COMPUTER LAW., June 1991, at 1, 2 (referring to user interface as a "dialogue between the user and the computer program"); Miller, *supra* note 13, at 348 (equating user interface with the "look and feel" of a software system).

[23] Reed, *supra* note 12, at 47.

ful software products to determine the underlying file formats and hardware interfaces, thus enabling the production of a competing substitute that is compatible with the commercially successful original version.[24]

In the recent case of *Lotus Development Corp. v. Paperback Software Int'l*,[25] a Massachusetts district court traced the competing views regarding the above issues in the context of computer spreadsheets. Although the producer of the popular spreadsheet Lotus 1-2-3 won the initial skirmish in the United States,[26] the war has yet to be won. The spreadsheet market is international. Hence, if Lotus Development Corp. (Lotus) attempts to enforce its rights in the Community, Lotus may face a different fight under the recently adopted Directive on the Legal Protection of Computer Programs.[27]

This Note attempts to foresee the analysis of *Lotus Development Corp. v. Paperback Software Int'l* under the newly adopted Directive. In Section II, this Note discusses the underlying purpose and lobbying efforts that led to the compromise Directive; it also analyzes the provisions of the Directive through a comparison to the United States Copyright Act. In Section III, this Note traces the history of United States copyright protection for computer programs that led to the *Lotus* decision. Finally, this Note analyzes the facts of *Lotus* under the Directive in an attempt to discover current trends in the international copyright protection of computer software.

II. Adoption of a Compromise Directive

A. *Stated Purpose of the Directive*

Unification of legal protection of computer programs within the member states prompted the proposal for the Directive.[28] Prior to the adop-

[24] *See id.* These competing substitute programs commonly are referred to as clones. *Id.*

[25] 740 F. Supp. 37 (D. Mass. 1990).

[26] *See id.* at 68-70.

[27] *See* Reed, *supra* note 12, at 50, 52.

[28] In its explanatory memorandum accompanying the proposal for the Directive, the Commission stated: "Divergences between the copyright statutes of the Member States as to the availability and scope of the protection [for computer programs] have caused the Commission to initiate the harmonization process in view of the objective of completing the internal market." Proposed Directive, *supra* note 5, 1989 O.J. (C 91) at 16. The preamble to the Final Directive reiterated this goal of harmonization:

> [C]ertain differences in the legal protection of computer programs offered by the laws of the Member States have direct and negative effects on the functioning of the common market as regards computer programs and such differences could well become greater as Member States introduce new legislation on this subject;

tion of the Directive, only a minority of the member states provided clear copyright protection for computer software,[29] and the length of this protection varied dramatically.[30] According to the explanatory memorandum accompanying the proposal for the Directive, these divergences in protection would affect not only the free circulation of software within the Community, but also might distort competition by influencing commercial marketing decisions,[31] thus having a direct effect on the EC's attempt to function as a single market. Furthermore, the Directive would promote the trend of standardization, thus improving legal certainty regarding the exclusive rights of copyrighted software.[32]

Another implicit concern prompting the proposal for the Directive was the seeming dominance of the United States in the software industry.[33]

> [E]xisting differences having such effects need to be removed and new ones prevented from arising, while differences not adversely affecting the functioning of the common market to a substantial degree need not be removed or prevented from arising;

Final Directive, *supra* note 10, 1991 O.J. (L 122) at 42.

[29] Of the 12 EC states, only the United Kingdom, West Germany, and France provide clear copyright protection for computer software. *EC Ministers Agree to Grant Protection for 50 Years for Copyrights of Software*, 7 Int'l Trade Rep. (BNA) 1923 (Dec. 19, 1990) [hereinafter *EC Ministers Grant Protection*].

Furthermore, in the Federal Republic of Germany, judicial decisions fixing a high level of originality as a threshold for copyright protection effectively have denied the majority of computer programs any protection under the copyright paradigm. *See* Thomas Dreier, *Program Protection in the Federal Republic of Germany—A New Decision Leaves Inkasso Programm Intact*, 7 COMPUTER L. & PRAC. 178, 179-80 (1991); M. Lehmann & Thomas Dreier, *The Legal Protection of Computer Programs: Certain Aspects of the Proposal for an (EC) Council Directive*, 6 COMPUTER L. & PRAC. 92, 92-93 (1990).

[30] For example, the length of copyright protection is generally at least 50 years, but France provided only 25 years of protection for computer programs. SCHROEDER, *supra* note 1, at 7.

[31] *See* Proposed Directive, *supra* note 5, para. 1.4, 1989 O.J. (C 91) at 5; *see also* SCHROEDER, *supra* note 1, at 12.

[32] *See* Final Directive, *supra* note 10, 1991 O.J. (L 122) at 43 ("[T]he Community is fully committed to the promotion of international standardization"); Proposed Directive, *supra* note 5, para. 3.9, 1989 O.J. (C 91) at 7 ("The provisions of this Directive should contribute to the trend towards a greater use of standardization."); *see also* SCHROEDER, *supra* note 1, at 13.

[33] *See* Proposed Directive, *supra* note 5, para. 1.3, 1989 O.J. (C 91) at 5. ("If the level of protection given to computer programs in Member States should fall below that accorded to programs created in other countries it is evident that the work of European innovators in this fast moving and highly competitive field will be easily appropriated by predatory activities from outside the Community."). The author of this Note suggests that the reference to predatory activities from outside the Community is an indirect assault on the United States dominance of the software field. *See* SCHROEDER, *supra* note

European software firms saw the Directive as a chance to provide a favorable legal environment in which to promote the development of competitive, compatible programs.[34] The Commission viewed the maintenance of uniform levels of protection as a means to stimulate research and investment in the EC's computer and technology industry in an attempt to cut into the United States market dominance of the software industry.[35] To stimulate research, the Commission stated that the Directive must prevent the works of Europeans from being appropriated by predatory activities outside the Community.[36] The initial proposal for the Directive did not contain an exception permitting reverse engineering.[37] This failure to provide an express provision on reverse engineering was viewed by many as a ban on the practice.[38]

B. *Precompromise Lobbying*

The silence on the permissibility of reverse engineering sparked intense lobbying efforts by groups representing diverse interests in the software industry. The two prominent groups involved in the lobbying process were the European Committee for Interoperable Systems

1, at 11-12. Because operating systems are often supplied with hardware, the EC views the software industry leaders within the United States (*e.g.* International Business Machines (IBM) and Digital Equipment Corp. (DEC)) as possessing a competitive advantage. *Id.*

[34] *See generally* SCHROEDER, *supra* note 1, at 11-13.

[35] *See* Proposed Directive, *supra* note 5, 1989 O.J. (C 91), at 5. "It is essential to create a legal environment which will afford a degree of protection against unauthorized reproduction . . . if research and investment in computer technology are to continue at a sufficient level to allow the Community to keep pace with other industrialized countries." *Id.* para. 1.3.

[36] *Id.* at 5. This purpose, however, has suffered criticism. One commentator urges that all industrialized states, not simply the EC, should have a high level of protection for computer programs. *See* SCHROEDER, *supra* note 1, at 12. "A low level of protection implies that European parties and non-Europeans alike can copy software developed outside of the European community. . . . Copying has never been the basis for long term growth and investment." *Id.*

[37] *See* Amended Proposal, *supra* note 10, 1990 O.J. (C 320) at 22-30 (side-by-side analysis comparing original proposal with amended proposal). The express provision authorizing decompilation when necessary for interoperability first appeared in the European Parliament's suggested amendments. *See* Parliament Amendments, *supra* note 10, 1990 O.J. (C 231) at 81.

[38] *See ECIS Position Statement*, *supra* note 14, at 99. *But see* A.B. Cleaver, *Reverse Engineering Could Be Misused*, FIN. TIMES, Sept. 17, 1990, at 19 (commentator urging the software industry to recognize that the proposal was merely silent on the issue of reverse engineering).

(ECIS)[39] and the Software Action Group for Europe (SAGE).[40]

ECIS, along with at least one economist, argued that a provision for reverse engineering was needed at least to enable competitors to determine the nonprotectible ideas underlying a successful software product.[41] Reverse engineering is a blanket term encompassing a large number of analytic techniques that permit a manufacturer to decompile or dismantle a competitor's program to produce compatible software or to build compatible equipment.[42] ECIS argued that established software markets are dominated by industry giants.[43] Only by producing products compatible with the market leaders can software newcomers, or small and medium sized companies, even hope to compete. Thus, ECIS argued that legitimate reverse engineering also should permit competitors to analyze a successful product to determine file structure and format in an effort to promote compatibility.[44]

In a position statement on the Commission's proposal for the Directive, ECIS argued that the proposal "should be revised by adding a new section . . . permitting reproduction and translation of a computer pro-

[39] The European Committee for Interoperable Systems (ECIS) represents small to middle sized software and hardware manufacturers, including Italy's Ing. C. Olivetti & Co., France's Groupe Bull, and Amdahl Corp. ECIS also represents IBM competitors such as Unisys Corp., Sun Microsystems, and NCR Corp. Elizabeth De Bony, *EC to Adopt Copyright Directive*, COMPUTERWORLD, Nov. 19, 1990, at 41.

[40] Sofware Action Group for Europe (SAGE), an alternative lobby representing 80% of the computer industry, is led by IBM and DEC. *Id.* Business Software Alliance (BSA), an active member of SAGE, represents business sofware industry leaders, including Lotus Development Corp. (Lotus), Microsoft, and Wordperfect. *European Parliament Acts on EC Software Dirctive*, Business Wire, July 12, 1990, *available in* LEXIS, Nexis library, Currnt file [hereinafter *Parliament Acts*].

[41] *See ECIS Position Statement, supra* note 14, at 100-01; W.R. Cornish, *Interoperable Systems and Copyright*, 11 EUR. INTELL. PROP. REV. 391 (1989); W.R. Cornish, *EC Directive on Programs*, FIN. TIMES, Mar. 15, 1990, at 23 (London School of Economics professor stating that the Directive should include an express exception permitting reverse engineering); *see also* Michel Colombe & Caroline Meyer, *Seeking Interoperability: An Industry Response*, 12 EUR. INTELL. PROP. REV. 79 (1990) (supporting Professor Cornish's views). *But see* William T. Lake et al., *Seeking Compatibility or Avoiding Development Costs? A Reply on Software Copyright in the EC*, 12 EUR. INTELL. PROP. REV. 431, 431-33 (rejecting Cornish's views). *See generally* Michael Becket, *Battlers Point to Enemy Within*, DAILY TELEGRAPH, Nov. 7, 1990, at 27 (industry leaders comment on proposed changes to the EC directive); De Bony, *supra* note 40, at 41 (France, Greece, and Germany also viewed a complete ban against reverse engineering as too restrictive).

[42] De Bony, *supra* note 39, at 41.

[43] *See* Becket, *supra* note 41, at 27.

[44] *ECIS Position Statement, supra* note 14, at 98.

gram for research and analysis purposes, in order to authorise [sic] expressly accepted industry practices."[45] This argument, which ultimately led to the recently adopted compromise Directive,[46] was logical and persuasive given the Commission's concerns as stated in the introductory remarks of the proposal for the Directive.[47] ECIS strongly argued that to prohibit reverse engineering for legitimate purposes, as opposed to piracy, would stunt the evolution of the computing industry.[48] Reverse engineering supposedly would promote competition and would keep the entire software industry on its toes. The consumer, therefore, would benefit because the software industry would evolve and improve without inflating prices.[49] This pro-competition reasoning by ECIS is probably the strongest indicator of why industry giants, such as International Business Machines (IBM) and Digital Equipment Corp. (DEC), favor strong copyright protection of computer software. Computer Users in Europe (CUE),[50] another lobbying group in favor of less restrictive copyright

[45] *Id.* at 101. The author of this Note suggests that this accepted practice of research and analysis can be equated with the practice of reverse engineering to discover the unprotectible ideas. ECIS proposed the addition of the following paragraph to article 5: "Reproduction and translation of a computer program, to the extent necessary to research, study or extract the unprotectable elements underlying the program shall not be restricted acts." *Id.*

[46] The European Parliament, on its first reading of the proposed Directive, suggested revision of article 5. *See* Parliament Amendments, *supra* note 10, 1990 O.J. (C 231) at 81 (adding a paragraph, similar to the one suggested by ECIS, to the article 5 exceptions); Final Directive, *supra* note 10, art. 5(3), 1991 O.J. (L 122) at 45 (essentially incorporating the European Parliament's suggestion). In addition to the article 5 modification, the Directive also contains an explicit decompilation provision. *See* Final Directive, *supra* note 10, art. 6, 1991 O.J. (L 122) at 45 (providing a decompilation exception when indispensible to obtain information necessary to achieve interoperability).

[47] *See* Proposed Directive, *supra* note 5, 1989 O.J. (C 91) at 5; *see also* SCHROEDER, *supra* note 1, at 11 (indicating goals of protecting software from unauthorized copying, meanwhile enabling small and medium sized enterprises to keep pace with other industrialized states).

[48] *See ECIS Position Statement, supra* note 14, at 100-101. SAGE, however, argued that reverse engineering would enable the "onslaught" of Japanese software to the detriment of the EC. John W. Verity, *Defense Against Pirates or Death to the Clones?*, BUS. WEEK, May 7, 1990, at 138, 140.

[49] *See* Becket, *supra* note 41, at 27. Clearly, under this theory, the only party that suffers is the software producer holding the copyright, which often will be an industry giant. *See id.* Furthermore, use of the word "suffer" is probably too strong because copyright law does not grant any entitlement to revenues from the unprotected, underlying ideas in publicly distributed software products. *See ECIS Position Statement, supra* note 14, at 101.

[50] Computer Users in Europe (CUE) includes large computer users, as opposed to software producers, such as Barclays Bank, Galileo International, the European airline

protection, was concerned that silence on the right to reverse engineer would thwart the development of "open systems."[51] CUE viewed the lobbying efforts of SAGE as "dominant [U.S.] computer companies seeking to convert their commercial positions into legal monopolies."[52]

SAGE, headed primarily by IBM and DEC, favored the strict protection found in the proposed Directive.[53] Advocates of this strong position argued that unlike hardware, software could be translated and duplicated (reverse engineered) quickly.[54] Thus, the lead time normally enjoyed by the original author would be reduced so greatly that the original author no longer would be able to recoup the research expenditure during the period of temporary monopoly provided by the lead time.[55] Because of this feared inability to recoup investment, SAGE apparently believed that lenient reverse engineering provisions would discourage the software industry from engaging in research and development.

Surprisingly, IBM was not alone in its support of strong protection. The formal bipartisan statement by the United States supported the strict protection with no mention of reverse engineering found in the Di-

reservation system, and the West German Aerospace Research Centre. Alan Cane, *Computer Users Fight EC Software Directive*, FIN. TIMES, Sept. 10, 1990, at 4.

[51] *Id.* Open systems "allow machines and software from different manufacturers to be used together." *Id.* CUE draws an analogy between compatible open systems and a hi fi system: when buying a stereo system, one can buy components from various companies. CUE argues that the same should be true for computer systems. *The Good of All: British Business Wakes Up to the Implications of a European Software Directive*, WHICH COMPUTER?, Nov. 1990, at 13 [hereinafter *The Good of All*].

[52] *The Good of All, supra* note 51, at 13.

[53] *EC Ministers Grant Protection, supra* note 29, at 1923.

[54] *See, e.g.*, Anthony L. Clapes et al., *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 UCLA L. REV. 1493, 1509 (1987). For example, the authors write:

> Software is different. There are typically no manufacturing processes to analyze, and no special factories to set up. Software is written and tested; it is then published, like books, records, or videotapes. It is possible to copy a computer program in seconds and readily reproduce that copy by the hundreds or thousands. It is more difficult, but nonetheless relatively easy, to adapt, translate, or "port" a program, and thereby appropriate much of the value inherent in the original author's creation. Software, by its nature, lends itself to quick and unexpected duplication and even translation.

Id. The views expressed by these authors have been criticized as being maxiprotectionist. *See* Samuelson & Glushko, *supra* note 21, at 135-36.

[55] *See* Clapes et al., *supra* note 54, at 1509; Lake et al., *supra* note 41, at 434 (copyright protection provides lead time required to earn return on "investment in creating an innovative program").

rective proposal.[56] United States officials apparently feared that an EC directive condoning reverse engineering would permit Japanese competitors to pirate software that had been developed in the United States, because those United States products making their way to the European market would be subject to reverse engineering under a lenient EC Directive.[57] This formal United States position, however, overlooks the objective of ensuring competitiveness of United States products on an international economy.[58] Protectionist intellectual property measures only offer short-term benefits to some, but not all, domestic vendors.[59] Those groups taking pro-protectionist stances fail to realize how quickly the software market and industry change—the dominant industry vendor at the present will not necessarily be the dominant force in the future.

This protectionist stance taken by SAGE has not passed uncriticized. Scholars have referred to this protectionism as an attempt by United States software giants to preserve "the last bastion ,of American technological superiority."[60] The potential problem with SAGE's protectionist view is that the software developer, whose product becomes the industry standard, can create a de facto monopoly.[61] This possibility of gaining a stronghold on the software market prompts vendors, such as IBM and Lotus, to compete fiercely for the initial sale. After this first sale, the user then becomes locked into products by this vendor unless compatible products are produced.[62] For example, in the context of a computer

[56] *Business Software Alliance Announces U.S. and European Companies Applaud U.S. Government Statement on EC Software Directive,* Business Wire, Feb. 7, 1990, *available in* LEXIS, Nexis Library, Currnt File. Thomas Niles, United States Ambassador to the EC, refuted assertions that United States copyright law permits reverse engineering. *Id. But see supra* part III.B (scholars finding support for reverse engineering in the Copyright Act).

[57] Andrew Hurst, *U.S. Fears Slacker EC Laws Will Let Japan Poach U.S. Software,* Reuter Business Report, Feb. 23, 1990, *available in* LEXIS, Nexis Library, Currnt File.

[58] *See* Michael A. Jacobs, *Copyright and Compatibility,* 30 JURIMETRICS J., 91, 104 (1989).

[59] *Id.* at 104.

[60] Reichman, *supra* note 3, at 695-96 (author referring to suggestion by Professor Karjala). Reichman urges Berne Union states to recognize the need for efficient allocation of copyrightable resources, especially when manufacturers of computer programs attempt to "avoid competition by masquerading as providers of cultural goods entitled to copyright protection on a par with literary and artistic works." J.H. Reichman, *Goldstein on Copyright Law: A Realist's Approach to a Technological Age,* 43 STAN. L. REV. 943, 948 (1991) (book review).

[61] *See* Joseph Farrell, *Standardization and Intellectual Property,* 30 JURIMETRICS J. 35, 38-39 (1989).

[62] *Id.* at 38. This phenomenon, recognized by the computer industry, has become

spreadsheet, a user who has invested time and effort to learn the particular spreadsheet interface faces an inertia against switching to another vendor's spreadsheet program with different interfaces.[63] Furthermore, the user will have accumulated spreadsheets containing masses of accounting and other critical data using the data structures and formats specified by the software developer. The cost re-entering or recompiling this data into a different format makes switching spreadsheet vendors economically unfeasible, unless some standardization of interfaces is permitted by copyright law.[64] Once this dependence on a single product manufacturer has been created, the lead or dominant vendor then can set price levels greater than cost without losing customers.[65] Even if compatibility becomes a stated policy goal, the lead vendor or standard leader still can gain significant competitive advantage by slightly changing the standard with little or no notice.[66] These slight changes suddenly make competitors' products no longer compatible; the industry second comer again finds itself struggling to maintain any market share that it had gained through the development of compatible components and software. Thus, this fear of a de facto monopoly and United States domination of the EC software market explains why ECIS and other lobbyists were able to achieve a compromise Directive permitting decompilation when indispensible to interoperability.[67]

C. *The Resulting Compromise Directive*

The lobbying efforts of ECIS and IBM's competitors resulted in a compromise on the issue of reverse engineering.[68] The revised Directive permits decompilation when "indispensible . . . to achieve the interoper-

known commonly as lock-in. *See* Colombe & Meyer, *supra* note 41, at 79-80; *see also* Thomas M. Hemnes, *Three Common Fallacies in the User Interface Copyright Debate*, 6 COMPUTER L. & PRAC. 163, 167 (recognizing the problem, but not explicitly referring to it as lock-in).

[63] *See* Hemnes, *supra* note 62, at 167; *see also* Jacobs, *supra* note 58, at 99 (using the example of a word processing system to emphasize the investment in training).

[64] *See* Colombe & Meyer, *supra* note 41, at 79-80.

[65] Farrell, *supra* note 61, at 38.

[66] *Id.* at 39-40. This fear has been seen in an EC antitrust suit against IBM. *See id.* at 40 n.12.

[67] *See* Final Directive, *supra* note 10, art. 6, 1991 O.J. (L 122) at 45 (explicit decompilation provision).

[68] *See* Final Directive, *supra* note 10, arts. 5, 6, 1991 O.J. (L 122) at 44-45. Article 5(3) arguably permits some level of reverse engineering not amounting to decompilation. Article 6 is an explicit decompilation exception to the author's exclusive rights.

ability of an independently created computer program."[69] The text of the Directive, however, does not contain the industry terms "decompilation" or "reverse engineering."[70] Nevertheless, article 6 of the Directive often is referred to as the decompilation provision. SAGE contends that the interoperability limitation should be clarified to safeguard against the use of decompilation to develop a competing, substitute product.[71] In contrast, ECIS believes that the provision should be read broadly to permit the development of a competing product, as long as the newly developed product is not a literal infringement[72] of the original.[73] By nonliteral infringement, the ECIS most likely means that unprotectible ideas and principles gleaned from the research and analysis process may be incorporated into a competing, yet independently created, software product.[74] Even under ECIS's broad interpretation, the compromise Directive, with its interoperability limitations on decompilation, may stifle innovation in the EC. Thus, the Directive still seems to benefit the dominant market players of the software industry—notably the industries of the United States and Japan.[75] This view, however, sees article 6, the decompilation

[69] *Id.* art. 6(1), 1991 O.J. (L 122) at 45. Interoperability is defined in the preamble of the Directive as "the ability to exchange information and mutually to use the information which has been exchanged." *Id.* at pmbl., 1991 O.J. (L 122) at 43.

[70] *See* Final Directive, *supra* note 10, arts. 5, 6, 1991 O.J. (L 122) at 44-45. The term "decompilation" appears as the title to article 6. The term, however, does not appear in the textual body of the exception. *Id.* For the full text of article 6, see *infra* note 127.

[71] *Parliament Acts, supra* note 40, *available in* LEXIS, Nexis library, Currnt file. According to a SAGE spokesperson:

> It is one thing to decompile an original program to develop a new interoperable program to attach to it. However, a broad right to use this process [decompilation] to try to replace that original program on the market would go well beyond the need for interoperability and would seriously jeopardize the ability of smaller firms to bring new and innovative products to market

Id. (SAGE comments on Parliament's amendment permitting decompilation when necessary for interoperability); *see also* Michel Colombe & Caroline Meyer, *Interoperability Still Threatened by EC Software Directive: A Status Report*, 9 EUR. INTELL. PROP. REV. 325, 327-29 (1990); *EC Ministers Grant Protection, supra* note 30, at 1923 (SAGE stating that it favored a restrictive reverse engineering provision).

[72] The ECIS broad interpretation of the provision suffers should European tribunals adopt the position of recent United States decisions protecting nonliteral aspects of computer programs. *See, e.g.*, Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990); Whelan Assocs. v. Jaslow Dental Lab., 797 F.2d 1222 (3d Cir. 1986).

[73] *EC Ministers Grant Protection, supra* note 29, at 1923; *see also* notes 41-44 and accompanying text.

[74] *See ECIS Position Statement, supra* note 14, at 101.

[75] Lucy Kellaway, *EC Directive Aims to Stop Software Piracy*, FIN. TIMES, Dec.

provision with its interoperability limitations, as the only provision under which reverse engineering possibly would be condoned. When analyzing the Directive, one must not forget the provision found in article 5, paragraph 3, which permits a user to observe, study, and test the program to determine the underlying, unprotected ideas and principles.[76]

1. Protectible Subject Matter and Exclusive Rights

Notwithstanding the controversy surrounding the compromise decompilation provision, the Final Directive, effective as of January 1, 1993,[77] substantially advances uniformity of laws and the protection of computer programs within the EC. Under the Directive, computer programs are protected by copyright as literary works.[78] The Directive does not define expressly the term "computer program."[79] The Directive, however, does state that the definition of computer program includes the preparatory design material.[80] This expanded definition of computer program ap-

13, 1990, at 22.

[76] *See* Final Directive, *supra* note 10, art. 5(3), 1991 O.J. (L 122) at 45. For a thorough discussion on the potential of article 5, see *infra* part II.C.2.a.

[77] Final Directive, *supra* note 10, art. 10, 1991 O.J. (L 122) at 46 ("Member States shall bring into force the laws, regulations and administrative provisions necessary to comply with this Directive before 1 January 1993.").

[78] Final Directive, *supra* note 10, art. 1(1), 1991 O.J. (L 122) at 44. Because this protection binds member states under the Berne Convention, protection of computer programs as literary works may be on the agenda for the next revision of the Berne Convention. SCHROEDER, *supra* note 1, at 14.

[79] *See* Final Directive, *supra* note 10, art. 1, 1991 O.J. (L 122) at 44. The preamble to the Directive, however, states: " '[C]omputer program' shall include programs in any form, including those which are incorporated into hardware; . . . this term also includes preparatory design work leading to the development of a computer program provided that the nature of the preparatory work is such that a computer program can result from it at a later stage;" Final Directive, *supra* note 5, 1991 O.J. (L 122) at 42. In contrast to the Directive, the Copyright Act avoids defining the term "computer program" in terms of itself: "A 'computer program' is 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.

The European Parliament apparently desired, but was unable to get, a more precise definition of computer program. The Parliament proposed an amendment defining the term "computer program" as "any sequence of instructions intended to be used, directly or indirectly, in a data processing system in order to carry out a function or obtain a specific result, independently of its form of expression." Parliament Amendments, *supra* note 10, 1990 O.J. (C 231) at 78. The innovative definition by Parliament even encompassed programs generated by the use of another program. *Id.*

[80] Final Directive, *supra* note 10, art. 1, 1991 O.J. (L 122) at 44 ("For the purposes of this Directive, the term 'computer programs' shall include their preparatory design material.").

pears to anticipate the protection of flow charts and may indicate a willingness to protect nonliteral elements of a computer program such as the "look and feel" of the user interface and the structure and organization of computer software. Nevertheless, the Directive adopts the idea/expression dichotomy even with respect to interfaces.[81] The explicit reference to the idea/expression dichotomy in the context of interfaces suggests that the EC may rule differently than some United States courts on the protection of "look and feel" if only the underlying ideas of the interface are copied and incorporated into a competing product. Such a different approach, however, may depend on how broadly EC tribunals read the interoperability limitation on decompilation.[82]

As in the United States Copyright Act, originality is the threshold for protection of a computer program in the Final Directive.[83] The Directive defines original as any work that constitutes an "author's own intellectual creation."[84] The preamble of the Directive also states that no qualitative or aesthetic test of merit should be applied to determine whether a computer program is an original work.[85] Thus, the EC appears to embrace the *Bleistein v. Donaldson Lithography Co.*[86] nondiscrimination theory found in United States copyright law,[87] as well as the view that independent creation satisfies the threshold originality requirement.[88]

[81] Final Directive, *supra* note 10, art. 1(2), 1991 O.J. (L 122) at 44 ("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."). *Cf.* 17 U.S.C. § 102(b) (United States law does not extend copyright protection to the underlying ideas).

[82] For a discussion of the interoperability limitation, see infra part II.C.2.b.

[83] Final Directive, *supra* note 10, art. 1(3), 1991 O.J. (L 122) at 44 ("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."). *Cf.* 17 U.S.C. § 102(a) (only original works of authorship are protectible subject matter under United States copyright law).

[84] Final Directive, *supra* note 10, art. 1(3), 1991 O.J. (L 122) at 44.

[85] *Id.* at pmbl., 1991 O.J. (L 122) at 42. This statement on originality further promotes uniformity because under pre-existing German law, courts had construed the originality threshold higher when determining protection of computer programs as opposed to traditional literary works. SCHROEDER, *supra* note 1, at 30; *see also* Dreier, *supra* note 30, at 178-79; Lehmann & Dreier, *supra* note 30, at 92-93.

[86] 188 U.S. 239 (1903).

[87] *See id.* at 251-52 (recognizing that judges and lawyers should not be making qualitative decisions regarding what works are deserving of copyright protection).

[88] *See id.* at 249-50; Alfred Bell & Co. v. Catalda Fine Arts, 191 F.2d 99, 102-03 (1951); *see also* 1 NIMMER & NIMMER, *supra* note 20, § 2.01[B] (discussing the originality threshold).

This view of originality rejects the substantial creativity test for originality,[89] even though the Directive, although not explicitly, seems to recognize the utilitarian or functional nature of computer programs.[90]

The exclusive rights provided to the copyright owner under the Directive[91] closely mimic the rights granted under the Copyright Act:[92] the exclusive right of reproduction,[93] the exclusive right to prepare derivative works,[94] and the exclusive right of distribution.[95] Nevertheless, as in the

[89] *See* L. Batlin & Son, Inc. v. Snyder, 536 F.2d 486, 490-91 (2d Cir. 1976) (requiring substantial creativity to satisfy the originality threshold for copyright protection of utilitarian works).

[90] *See* Final Directive, *supra* note 10, 1991 O.J. (L 122) at 43 (acknowledging in the preamble that computer programs serve the functions of communicating and working with other components of the computer systems and with users). The author of this Note argues that the use of the word "function" in the preamble of the Directive suggests that the EC has recognized implicitly the functional and utilitarian nature of computer programs. *But see ECIS Position Statement*, *supra* note 14, at 99-100 (criticizing the proposal of the Directive for failing to recognize explicitly the functional and utilitarian nature of computer programs). ECIS's criticism stemmed from the fear that copyright standards that have evolved in the literary context will be misapplied to computer programs. *Id.* at 100. The criticism, however, apparently went unheeded because the text of the Directive does not recognize explicitly the utilitarian nature of computer programs.

[91] Article 4 of the Final Directive provides:

> Subject to the provisions of Articles 5 and 6, the exclusive rights of the rightholder . . . shall include the 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, transmision [sic] 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, including the rental, of the original computer program or of copies thereof. The first sale in the Community of a copy of a program by the rightholder or with his consent shall exhaust the distribution right within the Community of that copy, with the exception of the right to control further rental of the program or a copy thereof.

Final Directive, *supra* note 10, art. 4, 1991 O.J. (L 122) at 44-45.

[92] *See* 17 U.S.C. § 106 (the exclusive rights of an author include reproduction, preparation of derivative works, distribution, performance, and display).

[93] *Compare* Final Directive, *supra* note 10, art. 4(a), 1991 O.J. (L 122) at 44 *with* 17 U.S.C. § 106(1) (both granting the exclusive right of reproduction).

[94] *Compare* Final Directive, *supra* note 10, art. 4(b), 1991 O.J. (L 122) at 44 *with* 17 U.S.C. § 106(2) (both granting the exclusive right to prepare derivative works).

[95] *Compare* Final Directive, *supra* note 10, art. 4(c), 1991 O.J. (L 122) at 44 *with* 17 U.S.C. § 106(3) (both granting the exclusive right of distribution).

Copyright Act,[96] the exclusive rights are subject to exceptions in the form of user rights.[97] Furthermore, the Directive provides that these exceptions cannot be overridden by a contractual provision.[98]

2. Exceptions to Exclusive Rights

a. Article 5—Observe, Study, and Test Exception

The exceptions provided in article 5 of the Directive[99] appear at first glance to be analogous to exceptions found in the Copyright Act, which expressly relate to computer programs.[100] The Directive permits the reproduction for the utilization of the computer program[101] and for archival purposes.[102] The Directive, however, extends beyond a mere duplication of section 117 of the Copyright Act. The Directive permits the observation, testing, and studying of a program to determine the underlying ideas and principles, if done while loading, displaying, running, transmitting, or storing the program.[103] Arguably, when article 5 is read

[96] *See* 17 U.S.C. §§ 107-118 (providing a milieu of exceptions to the § 106 exclusive rights, most notably, the fair use exception of § 107).

[97] *See* Final Directive, *supra* note 10, arts. 5, 6, 1991 O.J. (L 122) at 44-45 (providing for absolute exceptions, and an express decompilation exception if necessary for interoperability).

[98] *See* Final Directive, *supra* note 10, art. 9(1), 1991 O.J. (L 122) at 45 ("Any contractual provisions contrary to Article 6 or to the exceptions provided for in Article 5 (2) and (3) shall be null and void.").

[99] Article 5 of the Final Directive provides:

> 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 acquirer 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.

Final Directive, *supra* note 10, art. 5, 1991 O.J. (L 122) at 44-45.

[100] *See* 17 U.S.C. § 117 (limitations on exclusive rights: computer programs).

[101] *Compare* Final Directive, *supra* note 10, art. 5(1), 1991 O.J. (L 122) at 44 *with* 17 U.S.C. § 117(1) (both permitting reproduction when essential in the utilization of the computer program).

[102] *Compare* Final Directive, *supra* note 10, art. 5(2), 1991 O.J. (L 122) at 44 *with* 17 U.S.C. § 117(2) (both permitting reproduction for archival purposes).

[103] *See* Final Directive, *supra* note 10, art. 5(3), 1991 O.J. (L 122) at 45.

in conjunction with the article 4(b) right to prepare derivative works, the Directive does not permit reverse engineering simply to determine the underlying ideas and principles of a computer program because article 5 permits observation, but not copying or adaption.[104] Under this narrow view, the Directive lacks an exception analogous to the United States fair use exception.[105] This narrow interpretation, however, is flawed because the Directive does permit analysis (testing, studying, or observing) if done while loading, displaying, running, or storing the program.[106] Article 5, paragraph 3, permits this limited analysis if done while performing the specified acts of loading, displaying, running, transmitting, or storing the program—those acts that the user is entitled to do.[107] Article 5, paragraph 1, further permits the lawful acquirer of a computer program to reproduce the program (including reproduction for loading, displaying, running, transmitting, or storing) when necessary to use the program in accordance with its intended purpose.[108] Arguably, under article 5, paragraph 1, any user who has lawfully acquired a commercial software package would not infringe the copyright protection by loading and running the program because these acts are necessary for the utilization of the software. For example, a commercial spreadsheet software package is completely useless until it is loaded and run. Clearly, the loading and running would be acts permitted under article 5, paragraph 1, of the Directive.

In the computer industry, much research and analysis (both of which fall under the umbrella of reverse engineering) can be accomplished by loading the program into the computer memory, running it, and then viewing screen displays.[109] Thus, some analysis may be permitted under

[104] *See* Reed, *supra* note 12, at 52-53. The author, in supporting this narrow interpretation, argues even further that the user can observe only those things that are not protected by copyright. *See id.* at 53. This interpretation, however, is unduly narrow. The language of article 5(3) in no way restricts the observation to unprotectible elements. Article 5(3), however, does limit the purpose for observation—to determine the underlying ideas and principles. *See* Final Directive, *supra* note 10, art. 5(3), 1991 O.J. (L 122) at 45.

[105] *See* 17 U.S.C. § 107 (permitting reproduction of a copyrighted work for research purposes if the resulting use constitutes fair use). The following factors are used to determine whether the use made of the copyrighted work constitutes fair use: (1) whether the purpose of the use is of a commercial nature; (2) the nature of the copyrighted work; (3) the portion used in relation to the whole; and (4) the effect of the use upon the potential market for the copyrighted work. 17 U.S.C. § 107.

[106] *See* Final Directive, *supra* note 10, art. 5(3), 1991 O.J. (L 122) at 45.

[107] *See id.*

[108] *See id.* art. 5(1), 1991 O.J. (L 122) at 44.

[109] *See ECIS Position Statement, supra* note 14, at 100 (describing standard com-

article 5 through the legitimate running of the program in conjunction with research tools, such as a debugger.[110]

This broad interpretation of article 5 appears to be quite plausible, especially given the history leading to the adoption of article 5, paragraph 3. Article 5 of the proposed Directive contained only two paragraphs, which permitted reproduction necessary for utilization of the program and allowed use by the public in non-profit making libraries. The observe, study, and test exception found in paragraph 3 was not present at the proposal stage.[111] ECIS, after studying the proposed Directive, suggested an addition to the proposed article 5.[112] ECIS's article 5, paragraph 3, in essence, provided that reproduction by those users in proper possession of a program should be permitted when necessary for research and analysis to extract the unprotected elements of a program.[113] After its first reading of the proposal, the European Parliament

petitive analysis procedures); *see also* Vinje, *supra* note 10, at 4.

[110] A debugger is simply a computer program that, when run in conjunction with the application program to be analyzed, permits the computer software engineer to trace through the application program and observe how the program works. PETER NORTON & JOHN SOCHA, PETER NORTON'S ASSEMBLY LANGUAGE BOOK FOR THE IBM PC 5 (1986). For example, a debugger enables the sofware engineer to display the contents of registers and memory, to trace through the program being analyzed one line at a time, to execute portions of the program and stop before the execution of an instruction at a specified location, and to display status flags. *Id.* at 22, 35, 48. The debugger aids in the analysis process by translating the binary object code, which the machine executes directly, into hexadecimal code through which a trained software engineer is able to delve into the inner workings of a microprocessor as it executes an application program. *Id.* at 22, 55; *see also* Vinje, *supra* note 10, at 3 (referring to similar analysis techniques such as line traces, memory dumps, and screen displays of hexadecimal code).

The term "debug" originates from the fact that a working program has no disabling bugs or mistakes. According to folklore, the term "debug" stems from the failure of a computer at Harvard University. After searching for the source of the failure, the technicians discovered a "moth caught between the contacts of a relay. The technicians removed the moth and wrote a note in the log book about 'debugging' " the computer. NORTON & SOCHA, *supra*, at 5.

[111] *Compare* Proposed Directive, *supra* note 5, art. 5, 1989 O.J. (C 91) at 14 *with* Final Directive, *supra* note 10, art. 5, 1991 O.J. (L 122) at 44-45. *See also* Amended Proposal, *supra* note 10, 1990 O.J. (C 320) 22 (Commission document comparing original proposal with amended proposal for the Directive). Article 5(1) of the Directive remains essentially unchanged from the proposed Directive. The draft version of article 5(2) found in the proposed Directive (permitting use of the program by pubic libraries) is not present in the final version of article 5. The proposed public library exception now falls within the author's exclusive right of first sale. *See* Final Directive, *supra* note 10, arts. 4(c), 5, 1991 O.J. (L 122) at 44-45.

[112] *See ECIS Position Statement, supra* note 14, at 101.

[113] *Id.* The suggested amendment stated: "Reproduction and translation of a com-

appeared to take ECIS's suggestion to heart. The amended version of the Directive emerging from the European Parliament contained an additional paragraph, which allowed the legitimate holder of a program to observe, study, or test the working program in order to determine underlying ideas and unprotected elements when loading, viewing, running, transmitting, or storing the program.[114] Thus, the text of article 5, combined with the developments leading to its final adoption, supports a broad interpretation of article 5, paragraph 3.[115]

Nevertheless, article 5, paragraph 3, in its final form, contains the troublesome phrase: "any of the acts . . . which he [the user] is entitled to do."[116] If these acts are merely those necessary for the intended utilization of the program, the phrase is hardly troublesome.[117] If, however, these acts correspond to those specified in article 4(a), the phrase becomes a hurdle for advocates of reverse engineering. Article 4(a) provides that "[i]nsofar as loading, displaying, running, transmision [sic] or storage of the computer program necessitate such reproduction, such acts shall be subject to authorization by the rightholder."[118] If the acts of article 5, paragraph 3, are only those authorized by the rightholder under article 4(a), then any reverse engineering thought to be permitted

puter program, to the extent necessary to research, study or extract the unprotectable elements underlying the program shall not be restricted acts." *Id.*; *see also* Vinje, *supra* note 10, at 4 (stating that "moderates" in the industry first proposed article 5(3)); W.R. Cornish, *Computer Program Copyright and the Berne Convention*, 4 EUR. INTELL. PROP. REV. 129, 130-31 (1990) (arguing that ECIS's suggested amendment conformed with the Berne Convention).

[114] Parliament Amendments, *supra* note 10, 1990 O.J. (C 231) at 81. The paragraph added by Parliament stated:

> Notwithstanding the provisions of Article 4(a) [the exclusive right of reproduction], the legitimate holder 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.

Id.

[115] For support of this broad reading of article 5(3), see Vinje, *supra* note 10, at 3-4. *But see* Mindy J. Weichselbaum, Note, *The EEC Directive on the Legal Protection of Computer Programs and U.S. Copyright Law: Should Copyright Law Permit Reverse Engineering of Computer Programs?*, 14 FORDHAM INT'L L.J. 1027, 1040-41 (1991) (acknowledging article 5(3), but failing to recognize any potential authority therein for reverse engineering).

[116] Final Directive, *supra* note 10, art. 5(3), 1991 O.J. (L 122) at 45.

[117] *See* Vinje, *supra* note 10, at 4 ("[T]his provision is simply intended to guard against use of this article illegitimately to expand permitted uses of a program.").

[118] Final Directive, *supra* note 10, art. 4(a), 1991 O.J. (L 122) at 44.

under article 5, paragraph 3, would be illusory because most rightholders fearing competition would deny authorization of all acts, except those absolutely necessary for the utilization of the program. The Commission's explanatory memorandum accompanying the proposed Directive sheds some light on the above troublesome reading. In the comments on article 4(a), which underwent only minor change from the time of proposal until final adoption,[119] the memorandum states that " 'reproduction' should not be confused with 'replication.' "[120] According to this memorandum, recreation of the program in the computer memory during execution, or running, of the program constitutes replication, not reproduction, because no second permanent copy of the program is made during the process.[121] The moving and storing operations during execution leave no trace of the program once operation is terminated.[122] By including loading as a restricted act, or as one requiring the authorization of the rightholder, the Commission was anticipating future technology in which commercially available programs could be physically inserted into the computer and function as an integral part of hardware.[123] Under this scenario, reproduction or, more properly, replication in memory would not be necessary. Thus, if execution of the program is viewed as requiring merely replication of the program in memory, an act that the user is entitled to do, then the troublesome reading may be avoided.[124] Nonetheless, even without relying on some distinction be-

[119] *Compare* Proposed Directive, *supra* note 5, art. 4(a), 1989 O.J. (C 91) at 14 *with* Final Directive, *supra* note 10, art. 4(a), 1991 O.J. (L 122) at 44. *See also* Amended Proposal, *supra* note 10, 1990 O.J. (C 320) at 26. In the final version of the Directive, however, article 4(a) refers to "permanent or temporary reproduction," not simply "reproduction." Furthermore, the proposal considered loading, viewing, and running to be restricted acts. The Final Directive makes these acts merely subject to the authorization of the rightholder. Thus, article 4(a) seems to embody compromise. In exchange for the specific acts no longer being flatly restricted, reproduction now includes both permanent and temporary reproduction. *See* Final Directive, *supra* note 10, art. 4, 1991 O.J. (L 122) at 44.

[120] Proposed Directive, *supra* note 5, 1989 O.J. (C 91) at 10.

[121] *Id.*

[122] *Id.*

[123] *See id.* at 10-11.

[124] Reliance on the replication reference to avoid the potentially troublesome reading of article 5(3) is arguably misguided. Article 4(a) in the Final Directive refers to both permanent and temporary reproduction, which seems to include replication. *See* Final Directive, *supra* note 10, art. 4(a), 1991 O.J. (L 122) at 44. Nevertheless, a semantic argument can be made that replication is still distinguishable from temporary reproduction, especially given the Commission's anticipation of future technology in which programs function as an integral part of hardware. *See* Proposed Directive, *supra* note 5, 1989 O.J. (C 91) at 10-11.

tween replication and reproduction, the intended meaning of the "entitled to do" phrase is most likely that a user can observe, study, and test the program provided the acts of loading and running are necessary for its utilization. This nontroublesome reading is ultimately supported by an express statement in the Directive that the exception provided in article 5, paragraph 3, cannot be overridden by contract.[125] Hence, a strong argument can be made that article 5, paragraph 3, permits noninfringing reverse engineering that does not involve translation of the analyzed program's object code into "something akin to its original source code."[126]

b. Article 6—Explicit Decompilation Exception

Finally, the Directive explicitly recognizes a decompilation exception[127] when "indispensable to obtain the information necessary to

[125] *See* Final Directive, *supra* note 10, art. 9(1), 1991 O.J. (L 122) at 45; *see also* Vinje, *supra note* 10, at 4 (stating that a licensing provision dictating that a program be run solely for data processing and not for observing, studying, or testing would be ineffective).

[126] Vinje, *supra* note 10, at 3, 10 nn.23 & 38. For purposes of this Note, noninfringing reverse engineering refers to the use of analysis techniques, not rising to the level of decompilation, to glean unprotectible ideas for incorporation into an independently developed computer program. *See id.* at 3, 10 n.23 (distinguishing decompilation from other reverse engineering analysis techniques).

[127] Article 6, the decompilation exception, of the Final Directive provides:

> 1. The 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 the 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 development, 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

achieve the interoperability."[128] The prerequisite for falling within this exception to the author's exclusive right of reproduction include: (1) the party engaging in the decompilation must have a right to use the program being analyzed; (2) the information necessary must not have been previously made readily available; and (3) the decompilation must be confined to only those parts necessary for interoperability.[129]

The first of the prequisites is entirely reasonable. The remaining two are unduly restrictive. First, often the documentation provided by the software developer or copyright holder is incomplete.[130] Furthermore, simply because the documentation on the formats and interfaces necessary for interoperability is readily available, why then should software engineers automatically be deemed to engage in infringement simply by using decompilation techniques to discover the underlying ideas? Second, confining the analysis to only those parts of the program concerned with interoperability is not only overly restrictive, but also impractical. For example, if no documentation has been made available, clearly a software engineer must decompile the entire program to achieve interoperability of an independently created program. Furthermore, even if the software copyright holder were to specify exactly which parts of the program should be analyzed, few competent software engineers would be comfortable merely accepting this revelation without further analysis and study.

Even upon satisfying the conditions precedent for lawful decompilation, substantial limitations further restrict a user's ability to decompile programs.[131] The first limitation prohibits decompilation for any purpose other than achieving interoperability.[132] Again, this prohibition begs the question of why the act of decompilation should constitute infringement when used only to determine the underlying ideas, and not to copy protected expression. The article 6 limitations also prohibit publication of

> in such a way as to allow its application to be used in a manner which unreasonably prejudices the right holder's [sic] legitimate interests or conflicts with a normal exploitation of the computer program.

Final Directive, *supra* note 10, art. 6, 1991 O.J. (L 122) at 45.

[128] *Id.* art. 6(1), 1991 O.J. (L 122) at 45.

[129] *Id.*

[130] *See* Vinje, *supra* note 10, at 10, n.22 (recognizing the occasional need to reproduce dysfunctionality in a computer program; this dysfunctionality (or bugs) usually is not documented in manuals).

[131] *See* Final Directive, *supra* note 10, art. 6(1)(b), 1991 O.J. (L 122) at 45 (permitting decompilation provided that the information has not previously been made readily available).

[132] *Id.* art. 6(2)(a), 1991 O.J. (L 122) at 45.

information gained from decompilation, except when necessary for interoperability.[133] Most important, article 6, paragraph 2(c), does not permit decompilation for the development, production, or marketing of a computer program substantially similar to the protected expression of the original program.[134] This restriction on decompilation, however, turns on how EC tribunals will interpret the phrase "substantially similar" at the infringement stage.[135] If substantial similarity at the infringement stage is read broadly to include structure, sequence, and flow or even "look and feel," then this restriction would prohibit a software engineer from decompiling the program to develop a competing, interoperable product. Such a reading could be justified because the Directive prevents the application of any provision in a manner that would conflict with the rightholder's normal exploitation of the program.[136] This statement suggests that decompilation is not permissible if it will negatively affect the market interest of the original program.[137] This restrictive reading of the decompilation provision assumes that nonliteral elements of a computer program, such as the structure, sequence, and flow or "look and feel," constitute protected expression under the Directive. No one, however, confidently can make the assumption, because debate continues to rage over whether the copyright paradigm protects, or should protect, the nonliteral elements of a computer program.[138] If the Directive does broadly limit the decompilation exception, the compromise Directive is merely a hollow victory for ECIS. Nevertheless, article 5, paragraph 3, presents an untapped source of authority for reverse engineering techniques, not amounting to decompilation.[139]

The Directive has been praised for its laudable objectives that promote the "development of the Community market for information technology

[133] *Id.* art. 6(2)(b), 1991 O.J. (L 122) at 45.

[134] *Id.* art. 6(2)(c), 1991 O.J. (L 122) at 45.

[135] As stated by United Kingdom Trade Secretary John Redwood, even after the passage of the Directive, courts must determine the limits of reverse engineering. *See* De Bony, *supra* note 39, at 41.

[136] *See* Final Directive, *supra* note 10, art. 6(3), 1991 O.J. (L 122) at 45.

[137] *See, e.g.*, 17 U.S.C. § 107(4) (considering the effect of the use on the potential market in order to determine whether use of a work in a particular case constitutes fair use).

[138] *See, e.g.*, Clapes et al., *supra* note 54, at 1578-83 (seeming to embrace the view that design and other nonliteral elements result from the author's creative decisions and, hence, should be protected). *But see Samuelson & Glushko*, *supra* note 21, at 126-29 (reflecting the view of many industry professionals that the nonliteral elements of "look and feel" should not be protected).

[139] *See infra* notes 106-126 and accompanying text; Vinje, *supra* note 10, at 11 n.50.

products."[140] Nevertheless, a criticism of the Directive is that although it recognizes the functionality of software, computer programs are protected as literary works.[141] This imperfect analogy to literary works fails to recognize that the source code (and object code) is but one aspect of a software system.[142] A functional product "unlike a book . . . needs to be copied just to use it to see what it does."[143] Moreover, reverse engineering, including decompilation, to produce a competing product technically may infringe on the original author's exclusive right of reproduction, but the final competing product may not be substantially similar in expression to the copyrighted product, especially if only ideas are incorporated into the independently developed competing program. Under the Directive, reverse engineering for purposes other than achieving interoperability depends on how broadly the article 4(b) right to prepare derivative works extends.[144] Furthermore, even if EC tribunals interpret article 5, paragraph 3, as permitting some reverse engineering unrelated to interoperability, software engineers then face a difficult evidentiary problem in an infringement suit—how to prove that the analytic techniques employed did not amount to decompilation. Finally, if all reverse engineering is subject to the interoperability limitation, competitors may be forced to fabricate some sort of interoperability excuse to fall within the Directive's explicit decompilation exception.

III. Status of United States Legal Protection of Computer Programs

Because the market for computer software is international in scope, the Directive's impact will extend beyond the EC. The Directive potentially will have direct consequences on the United States software industry, especially if industry leaders such as Lotus attempt to enforce the exclusive right of reproduction granted under the copyright paradigm in

[140] Miller, *supra* note 14, at 350.

[141] *Id.* at 348-49. The same defect also can be found in the analysis of United States protection of computer programs. In the Copyright Act, literary works are defined as "works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects." 17 U.S.C. § 101. Clearly both source and object code fall within this definition.

[142] Miller, *supra* note 14, at 349.

[143] *Id.* Contrast the case of a three-dimensional product in which examples may be bought, taken apart, and tested without infringing the producer's rights. *Id.*

[144] Article 1(2) of the Directive suggests that the author's right to prepare a derivative works should be interpreted narrowly, or at least in a manner that does not extend protection to the underlying ideas. *See* Final Directive, *supra* note 10, art. 1(2), 1991 O.J. (L 122) at 44.

other jurisdictions. Hence, this Note now takes a brief look at the evolving legal protection of computer programs in the United States. Following the synthesis of United States law, the Note concludes by analyzing a controversial "look and feel" infringement suit under the newly adopted Directive for the purposes of comparing United States law with the newly afforded EC protection.

A. *Evolving Scope of Protection for Computer Software*

The EC is not unique in its debate over the legitimacy of reverse engineering. Copyright scholars in the United States also differ as to whether domestic law permits reverse engineering for competition and interoperability purposes.[145] The debate in the United States is complicated further because protection of computer software is afforded under a general copyright statute originally designed for the protection of literary and artistic works, rather than a statute, like the Directive, specifically designed for the protection of computer software.[146] In the United States, the reverse engineering debate is spawned from the more fundamental issue of what is the proper scope of protection for computer programs under the copyright paradigm—whether copyright protection merely guards against the duplication of literal text or whether it extends to paraphrases, translations, and nonliteral copying.[147]

In determining whether nonliteral copying constitutes infringement, United States courts generally focus on the idea/expression dichotomy.[148] For example, in *Whelan Assocs. v. Jaslow Dental Lab.*,[149] the United States Court of Appeals for the Third Circuit found infringement of the nonliteral aspects of a computer program—the structure, flow, and organization of the program.[150] In determining infringement, the *Whelan* court rejected the bifurcated approach to finding infringement, which appeared in *Arnstein v. Porter*,[151] and instead adopted a single-step sub-

[145] *See infra* part III.B.

[146] In 1980, Congress amended the Copyright Act to recognize that computer programs are to be protected under the Copyright Act as literary works. 1980 Amendments to Patent and Trademark Law, Pub. L. No. 96-517, sec. 10, §§ 101, 117, 94 Stat. 3015, 3028 (1980); *see also* Clapes et al., *supra* note 54, at 1497 n.4.

[147] *See* Clapes et al., *supra* note 54, at 1502-04.

[148] *See* Reed, *supra* note 12, at 48; *see also* 17 U.S.C. § 102(b) (codification of idea/expression dichotomy).

[149] 797 F.2d 1222 (3d Cir. 1986).

[150] *Id.* at 1248.

[151] 154 F.2d 464 (2d Cir. 1946). Under the bifurcated approach, infringement is proven by first showing substantial similarity and access, followed by a showing that protectible expression (not § 102(b) matter) was taken. *Id.* at 468-69.

stantial similarity test for infringement.[152] The *Whelan* decision to apply a single-step substantial similarity test arguably goes beyond the protection of expression and thus grants protection for ideas that supposedly are unprotectible because the two programs are compared for substantial similarity without any concern for whether the substantial similarity results merely from a similarity of unprotectible ideas, not expression.[153] In contrast, the *Arnstein* bifurcated approach avoids protection of underlying ideas, yet still defends the author's market interest.[154] Hence, the *Whelan* decision, which protected the structure, sequence, and flow of a computer program after applying a single-step substantial similarity test, can be criticized as overprotecting computer programs.[155] Nevertheless, the argument may be made that detailed design, structure, and organization is not dictated by section 102(b) elements, but rather results from creative decisions.[156]

Further complicating the issue of protection is the dispute over what level of originality in computer programs is required to meet the threshold of copyright protection. In traditional artistic works, the originality requirement is met merely by showing independent creation.[157] But with utilitarian works, some courts require a stronger showing of quantitative creativity because independent creation may have little to do with expression of an author's personality.[158] With utilitarian products, market value and independently created elements often are dictated by external technological constraints, not by personal expression.[159] Thus, at least

[152] *Whelan*, 797 F.2d at 1233.

[153] *See* 3 NIMMER & NIMMER, *supra* note 20, § 13.03[A][1][d], 13-38 to 13-40 (suggesting that *Whelan* was correctly decided on the facts, but the sweeping language extends copyright protection too far).

[154] *See Reichman, supra* note 60, at 956 & n.89.

[155] *See, e.g.*, Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., 807 F.2d 1256, 1262 (5th Cir. 1987) (declining to accept *Whelan*'s broad holding that the structure, sequence, and organization of computer programs are copyrightable). The court in *Plains Cotton* did not flatly reject *Whelan*, but rather determined that on the facts presented, the sequence and organization had been dictated by external constraints. *Id.*

[156] *See* Clapes et al., *supra* note 54, at 1580 (arguing that detailed design and flow are not dictated by functionality); Johnston & Grogan, *supra* note 22, at 2-3 (stating that design of user interface requires original authorship by software developers).

[157] *See, e.g.*, Bleistein v. Donaldson Lithography Co., 188 U.S. 239, 243-44 (1903); Alfred Bell & Co. v. Catalda Fine Arts, 191 F.2d 99, 102-03 (1951); *see also* 1 NIMMER & NIMMER, *supra note* 20, § 2.01[B], at 2-11 to 2-13; Reichman, *supra* note 3, at 684.

[158] Reichman, *supra* note 3, at 684; *see, e.g.*, L. Batlin & Son, Inc. v. Snyder, 536 F.2d 486, 490-92 (2d Cir. 1976); Durham Indus. v. Tomy Corp., 630 F.2d 905, 909-11 (2d Cir. 1980).

[159] *See* Reichman, *supra* note 60, at 953-54 & n.62.

one scholar argues that computer programs should be viewed as "works of applied literature," requiring quantitative creativity to establish originality.[160]

B. *The Reverse Engineering Debate Under the Copyright Act*

Under United States law, independent creation is a perfect defense to copyright infringement.[161] Yet this defense is unrealistic in a field such as computer science in which "innovation occurs through sequential and cumulative improvements."[162] The need for computer scientists to borrow from pre-existing advancements suggests that certain forms of reverse engineering should be permitted in the software industry.[163] This need for reverse engineering finds support under the presumption that copying is an integral step in the advancement of knowledge.[164] The activity of reverse engineering is not per se improper because it has been condoned in both the patent paradigm[165] and in the sui generis protection of semiconductor chips.[166] Several leading commentators find support for the process of reverse engineering under the current United States copyright law. Each commentator, however, differs as to which doctrinal foundations actually support the practice.[167]

Reverse engineering under the Copyright Act primarily is justified by drawing a distinction between legitimate competition and piracy.[168] Copyright infringement is not necessarily a consequence of reverse engineering,[169] as when the newly developed product incorporates merely the ideas gleaned from the original software product. Hence, at least one United States Court of Appeals and several leading commentators support a reverse engineering exception to the exclusive rights afforded to a

[160] Reichman, *supra* note 3, at 688.

[161] *Id.* at 689.

[162] *Id.* (quoting Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 JURIMETRICS J. 33, 39 (1987).

[163] *See id.* at 689-90.

[164] BENJAMIN KAPLAN, AN UNHURRIED VIEW OF COPYRIGHT 2 (1966); *see* Leo J. Raskind, *Reverse Engineering, Unfair Competition, and Fair Use*, 70 MINN. L. REV. 385, 387 n.10 (1985).

[165] *See* Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 156, 160-61 (1989) (reverse engineering using unpatented technology promotes the goals of the patent system).

[166] *See* 17 U.S.C. § 906 (the Semiconductor Chip Protection Act of 1984 (the Chip Act) permitting reverse engineering).

[167] *See supra* text and accompanying notes 168-188, 192.

[168] Raskind, *supra* note 164, at 389.

[169] Reichman, *supra* note 3, at 702.

software author.

In *Vault Corp. v. Quaid Software Ltd.*,[170] the United States Court of Appeals for the Fifth Circuit has justified reproduction for the purpose of reverse engineering under the section 117 explicit exception[171] to the exclusive rights afforded to a software copyright owner.[172] The analysis of *Vault Corp. v. Quaid Software, Ltd.*, however, is probably a strained reading under both the United States copyright law and the EC Directive because the section 117 exception and the analogous exception under the Directive are limited to archival purposes and the use of the original program.[173]

Professor Paul Goldstein also argues for a reverse engineering limitation on the copyright protection afforded to computer programs.[174] Goldstein finds support for his theory under the fair use exception, found in section 107 of the Copyright Act.[175] Goldstein's theory argues that computer programs, unlike traditional copyrighted works, do not bear unprotected ideas on their face.[176] Furthermore, because vendors distribute software products in object code,[177] the software must be decompiled into human-readable form to permit analysis of the underlying, unprotected elements.[178] Technically, this decompilation constitutes infringement of the section 106(1) exclusive right of reproduction. Goldstein, however, finds nothing unlawful about "managed copying."[179] Under Goldstein's fair use approach, making a copy to design a competing, but noninfringing, product should be deemed "research," and thus fair use under section 107.[180] The commercial exploitation of the competing product should not automatically convert the practice of reverse engineering into

[170] 847 F.2d 255 (5th Cir. 1988).

[171] *See* 17 U.S.C. § 117 (providing that reproduction of the computer program is not infringement when the copy is needed as "an essential step in the utilization of the computer program," or for archival purposes).

[172] *See Vault*, 847 F.2d at 261.

[173] *See* 17 U.S.C. § 117; Final Directive, *supra* note 10, arts. 5(1) and (2), 1991 O.J. (L 122) at 44.

[174] *See* PAUL GOLDSTEIN, COPYRIGHT: PRINCIPLES, LAW AND PRACTICE § 5.2.1.4, at 85-91 (1991 Supp.).

[175] GOLDSTEIN, *supra* note 174, at 88-91.

[176] *Id.* at 86.

[177] *Id.* Object code is incomprehensible to humans. It is, however, machine executable. *Id.*

[178] *Id.*

[179] *Id.* at 86-87. Goldstein defines managed copying as a study of the lawfully obtained copyrighted software to extract unprotectible elements and to incorporate those elements into a competing work. *Id.* at 86.

[180] *Id.* at 89.

a prohibited act. Instead, infringement should be found only if the competing work copies the expression of the original software.[181] Goldstein's fair use justification of reverse engineering has been deemed a "public benefit" approach because the technical copying or reproduction should be excused if the social benefit of the use outweighs the loss to the copyright owner.[182]

Goldstein's view of United States copyright law arguably finds no analogous support under the Directive because article 5 appears only to permit observation, not copying for commercial purposes.[183] Also, the Directive's interoperability limitation on decompilation does not sweep as broadly as Goldstein's theory[184] because Goldstein would permit decompilation even when it is not indispensible to interoperability.[185] Nevertheless, contrary to SAGE's opinion, competing software arguably could be produced if a legitimate interoperability concern exists, assuming, as Goldstein does, that only ideas and not the expression of the original software have been included in the competing product.[186] Furthermore, if article 5, paragraph 3, of the Directive is interpreted broadly to permit some reverse engineering unrelated to interoperability;[187] then the Directive arguably supports Professor Goldstein's fair use rationale because the Directive would permit competitors to analyze (observe, study, and test) a commercial software package with the goal of extracting the underlying ideas and unprotected elements for incorporation into a competing substitute program. The Directive, however, even with this broad interpretation, does not venture as far as Goldstein because decompilation would constitute infringement unless a legitimate interoperability concern existed.[188]

In contrast, the renowned author Melville Nimmer never acknowledged the permissibility of reverse engineering under the Copyright Act.[189] Professor Nimmer supported a "design" or "pattern" theory for

[181] *Id.* at 89, 91.

[182] *See* Reichman, *supra* note 60, at 960-61.

[183] *See* Final Directive, *supra* note 10, art. 5, 1991 O.J. (L 122) at 44-45.

[184] *See id.* art. 6, 1991 O.J. (L 122) at 45.

[185] *See* GOLDSTEIN, *supra* note 174, § 5.2.1.4, at 87-91.

[186] But this interpretation of the Directive depends on whether EC tribunals will interpret "substantially similar in its expression" according to the *Arnstein* bifurcated analysis. *See supra* notes 151-54 and accompanying text for a discussion of the *Arnstein* analysis.

[187] For a complete discussion of this broad interpretation, see *supra* notes 106-26 and accompanying text.

[188] *See* Final Directive, *supra* note 10, art. 6, 1991 O.J. (L 122) at 45.

[189] *See generally* 3 NIMMER & NIMMER, *supra* note 20, § 13.03.

analyzing software infringement.[190] Under this "pattern" theory, infringement results from the appropriation of a program's organization—any "comprehensive nonliteral similarity" triggers a finding of copyright infringement.[191] The Directive could be construed as supporting Nimmer's restrictive view of reverse engineering, provided that the EC tribunals interpret the "substantial similarity" of article 6 according to the "pattern" test.

Obviously, the EC courts will be the ultimate arbitors of how broadly the interoperability requirement restricts the practice of reverse engineering to develop a competing product and whether the article 5 observe, study, and test exception permits any reverse engineering unrelated to interoperability. Yet if the EC courts follow the trend of United States copyright scholars,[192] the Directive probably will be construed broadly as advocated by the ECIS.[193]

C. *United States Concern for Interoperability*

As in the EC Directive, United States commentators recognize the need for the promotion of compatibility and interoperability.[194] The argument in favor of reverse engineering based on a compatibility or interoperability rationale proceeds as follows. With computer software, and especially in the case of operating system software, external constraints often dictate conformity with industry standards of format and expression.[195] When external constraints dictate expression, idea and expression have merged, and thus the expression should not be protected.[196]

[190] *Id.* § 13.03[A][1][b]; *see also* Clapes et al., *supra* note 54, at 1550 (citation omitted).

[191] *See* 3 NIMMER & NIMMER, *supra note* 30, § 13.03[A][1][b]; *see also* Clapes et al., *supra* note 56, at 1550 (citation omitted). *But see* 3 NIMMER & NIMMER, *supra* note 20, at § 13.03[F] (recent edition by David Nimmer advocating a "successive filtering" test).

[192] Goldstein is not alone in his pro-reverse engineering stance. *See* Reichman, *supra* note 3, at 692-93 & n.288. (author interprets *Baker v. Selden* as permitting reverse engineering to allow other authors to use functional features embodied in utilitarian works); Pamela Samuelson, *Creating a New Kind of Intellectual Property: Applying the Lessons of the Chip Law to Computer Programs*, 70 MINN. L. REV. 471, 524-25 (1985) (author advocates the adoption of a sui generis scheme for the protection of computer programs analogous to the Chip Act, which permits reverse engineering).

[193] *See supra* text and accompanying notes 72-74.

[194] *See, e.g.*, Farrell, *supra* note 61, at 43-44; Jacobs, *supra* note 58, at 92, 99-100.

[195] *See* Clapes et al., *supra* note 54, at 1536 & n.158, 1540 & n.170 (acknowledging, but rejecting, the arguments that the requirements of operating system programs and compatibility concerns often dictate expression).

[196] *See supra* note 20; *see also* Farrell, *supra* note 61, at 47; Clapes et al., *supra*

When merger results, reverse engineering should be condoned to allow others to discover these unprotectible elements.[197] Furthermore, because compatibility[198] and standardization[199] make economic sense,[200] simple economics dictate reverse engineering. Decompilation or reverse engineering for compatibility and interoperability purposes brings new products to the market and promotes innovation without asking customers to sacrifice any previous investment in data, software, and training.[201] The following network externalities also result from the standardization of certain aspects of computer products, which was an underlying purpose in the proposal for the Directive:[202] (1) the facilitation of computer networks; (2) the transfer of files among users and across applications; (3) the reduction in training costs; and (4) the increase of competition and innovation because now competitors can focus on the development of an individualized component, not the development of an entire system.[203] The only perceived costs of industry standardization are the possible loss of consumer choice or variety and the retardation of innovation that a standard may induce.[204]

On the other side of the interoperability debate, the maxiprotectionist view claims that reverse engineering advocates must never lose sight of the distinction between necessity and convenience: seeing the same screen format in competing products is convenient, but not necessary.[205] Maxiprotectionists believe that software clones and compatibility can be achieved without reverse engineering and without the copying of

note 54, at 1540.

[197] *See* Farrell, *supra* note 61, at 47 (suggesting user interface, format for data storage and transmission, and other rather arbitrary aspects should be unprotected to encourage standardization).

[198] Compatibility merely means that a competing vendor can develop software that runs easily with existing software, connects to existing hardware, and uses existing data files. Jacobs, *supra* note 58, at 99; *see also* Farrell, *supra* note 61, at 36.

[199] Standardization simply entails making products similar enough to be compatible. *See* Farrell, *supra* note 61, at 36.

[200] *See* Farrell, *supra* note 61, at 36.

[201] Jacob, *supra* note 58, at 100; *see also* Farrell, *supra note* 61, at 36-37; *supra* notes 63-68 and accompanying text.

[202] *See* Proposed Directive, *supra note* 5, 1989 O.J. (C 91) at 7 (recognizing a trend towards standardization of computer products).

[203] *See* Farrell, *supra* note 61, at 36.

[204] *Id.* at 36-37.

[205] *See* Clapes et al., *supra* note 54, at 1566 (stating that sparing computer users from the need to learn new screen formats or command terms is not necessary for compatibility).

expression.[206]

Yet this maxiprotectionist view begs the question of what purpose competition serves if the original protected expression was dictated by efficiency concerns[207]—what commercial success could an inefficient clone possibly enjoy? The industry leaders seem to be overly concerned with a free rider problem in an effort to conceal their ulterior motive of protecting their lead-time advantage by forcing competitors to reinvent an inefficient wheel.[208] But contrary to the fears of Lotus and other SAGE members, reverse engineering for purposes of standardization and compatibility will not destroy the benefits afforded to the first developer because the industry leader will still enjoy lead time, albeit shortened, as well as the reputation of being the first innovator on the market.[209]

IV. ANALYSIS OF A "LOOK AND FEEL" INFRINGEMENT SUIT UNDER THE DIRECTIVE

Lotus, as the developer of the commercially successful spreadsheet program Lotus 1-2-3, embraces the maxiprotectionist SAGE position that competing compatible programs can be developed without appropriating the user interface of industry leaders' software.[210] In contrast, second comers to the spreadsheet market wish to create competing products that will be compatible with the Lotus-created spreadsheets. This compatibility would enable the transfer of spreadsheet files between competing software products without loss of functionality of any macros in the spreadsheet.[211] ECIS and industry second comers fear that strong protec-

[206] *Id.* at 1566 & n.267 (citing cases in which the second comer has conformed its competing program or clone to formats imposed by external constaints, without copying the expession of the original program); *see also* Weichselbaum, *supra* note 115, at 1065-66 (stating that decompilation is not the best way of ensuring interoperability).

[207] *See, e.g.,* Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., 807 F.2d 1256, 1262 (5th Cir. 1987). The efficiency concerns involved in *Plains Cotton* were the externalities of the cotton market that dictated the structure and organization of input and output formats for cotton marketing sofware. *Id.; see also* Clapes et al., *supra* note 54, at 1567 n.267.

[208] *See* Jacobs, *supra* note 58, at 102 (arguing that independent development of compatible products does not introduce a "free rider problem").

[209] *See id.* at 104.

[210] Evidence of this maxiprotectionist stance is demonstrated by the frequency with which Lotus opts for litigation to curb the appropriation of its spreadsheet interface by competing software rivals. *See* William Rodarmor, *Litigating Not Innovating*, CALIFORNIA LAW., Mar. 1992, at 36, 36 (citing the initiation of lawsuits against Mosaic Software, Paperback Software Int'l, and, most recently, Borland Int'l); *see also* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990).

[211] *See* Michael Becket, *Paradox That's Often Settled in Court*, DAILY TELE-

tion of the "look and feel" of user interfaces combined with limited reverse engineering and decompilation rights will convert the EC into merely a marketplace for non-Community products manufactured by United States industry leaders, such as Lotus and IBM.[212]

A. Lotus Development Corp. v. Paperback Software Int'l

In *Lotus Development Corp. v. Paperback Software Int'l*,[213] the defendant attempted to attack the stronghold that Lotus had gained in the computerized spreadsheet industry. The defendant's competing spreadsheet VP-Planner mimicked the menu structure and the arrangement of menu commands allegedly to achieve compatibility with the industry giant and standard-dictating Lotus 1-2-3.[214] The defendants went so far as to publicly advertise the VP-Planner as a "workalike for 1-2-3."[215] The district court ruled that VP-Planner infringed both the menu structure and the macro facility of Lotus 1-2-3.[216] The basic screen displays and the choice of function keys used to invoke the menu system, however, were found not to infringe Lotus 1-2-3.[217] The district court applied a three part test to determine infringement: (1) apply the idea/expression dichotomy to the work seeking legal protection; (2) if not an unprotectible idea, determine if the expression of the underlying idea is limited so as to invoke the merger doctrine; and (3) after identifying the elements of expression, focus on substantial similarity.[218] After applying this test the district court determined that only the menu structure and macro facility had been unlawfully appropriated. The screen displays and the function keys, however, were capable of being expressed in only a limited number of ways, and hence the merger doctrine precluded a finding of infringement.[219]

GRAPH, Jan. 14, 1991, at 23.

[212] *See* Mark Hamilton, *Directive Threatens EC Computer Industry*, COMPUTERWORLD, Nov. 12, 1990, at 118.

[213] 740 F. Supp. 37 (D. Mass. 1990). For greater in-depth discussion of the *Lotus* decision, see generally Joseph T. Verdesca, Jr., Comment, *Copyrighting the User Interface: Too Much Protection?*, 45 S.w. L.J. 1047, 1066-74 (1991); Thomas K. Pratt, Comment, *A Legal Test for the Copyrightability of a Computer Program's User Interface*, 39 KANSAS L. REV. 1045 (1991).

[214] *Id.* at 68-69.

[215] *Id.* at 69.

[216] *Id.* at 62-70.

[217] *Id.* at 66.

[218] *Id.* at 60-61.

[219] *See id.* at 66-67.

B. *The "Look and Feel" Debate*

Nevertheless, by finding infringement of the menu structure and the macro facility, the *Lotus* district court, in effect, was endorsing the protection of the "look and feel" of user interfaces.[220] "Look and feel" refers to those "aspects of a program that a user sees when a computer is operating under the control of that program."[221] Arguably, "look and feel" lawsuits could be described as the third wave of computer infringement actions (the first wave involved slavish copying; the second wave involved nonliteral copying of elements such as the structure, sequence, and organization).[222] This "look and feel" concept as applied to computer programs has evolved from the protection of the "total concept and feel" of standard literary and artistic works.[223] Some commentators, however, believe that the protection of the "look and feel" of functional works, such as computer software, would grant industry first comers a de facto monopoly on unprotectible ideas.[224] Maxiprotectionists, nonetheless, argue that the nondiscrimination principle dictates that computer software should be accorded the same protection available for all other works of the imagination.[225] Some maxiprotectionists go so far as to argue that the "look and feel" of software involves more creativity and is of greater commercial value than the actual source code.[226]

This maxiprotectionist view, however, naively fails to realize that functional works, such as computer programs, trigger the "two-market conundrum" in which copyright protection also has an effect on the products market.[227] In this two-market conundrum, the exclusive rights

[220] For a general discussion of "look and feel" debate, see Samuelson & Glushko, *supra* note 21, at 121-23.

[221] Clapes et al., *supra* note 54, at 1503 n.29.

[222] *Id.* at 1502-03; *see, e.g.,* Whelan Assocs. v. Jaslow Dental Lab., 797 F.2d 1222 (3d Cir. 1986) (example of second wave case in which the court extended copyright protection to the structure, sequence, and organization of a computer program).

[223] *See, e.g.,* Roth Greeting Cards v. United Card Co., 429 F.2d 1106, 1110 (9th Cir. 1970) (referring to the "total concept and feel" of greeting cards); Reyher v. Children's Television Workshop, 533 F.2d 87, 92 (2d Cir.), *cert. denied,* 429 U.S. 980 (1976) (referring to the overall mood and detail of a book); *see also* 3 NIMMER & NIMMER, *supra* note 20, § 13.03[4][1][c] (describing the evolution of the "look and feel" concept as it applies to computer programs).

[224] *See* Clapes et. al., *supra* note 54, at 1503.

[225] *See id.* at 1504.

[226] *See* Elizabeth Ranney, *'Look and Feel' Discussed as Major Copyright Issue,* INFOWORLD, Nov. 11, 1985, at 13.

[227] J.H. Reichman, *Design Protection in Domestic and Foreign Copyright Law: From the Berne Revision of 1948 to the Copyright Act of 1976,* 1983 DUKE L.J. 1143, 1197.

may provide first comers with claims to nonprotectible matter, and a claim can then be used to harass potential competitors with threats of infringement suits.[228] Hence, the rightholder of the industry standard stands to reap a huge reward if the protection of "look and feel" is combined with a restrictive view on the legitimacy of reverse engineering and decompilation.[229] Because of this opportunity to reap the reward of a de facto monopoly, once a user interface, which is merely a format for data, becomes standard in the industry, it should remain unprotected.[230] As seen in *Kepner-Tregoe, Inc. v. Carabio,*[231] courts must be sensitive to any overwhelming market power of the plaintiff in a software infringement suit;[232] courts must not forget to balance the author's right against the public interest in competition.[233]

C. *The Facts of* Lotus *Analyzed under the Directive*

The desire to preserve strong market interests prompted IBM and Lotus to lobby fiercely for maxiprotectionist measures in the EC Directive. The outcome of a *Lotus*-type infringement suit governed by the Directive remains to be seen. The EC, however, will be engaging in this analysis should Lotus decide to bring an infringement suit in the Community after the effective date of the Directive. As to additional judicial analysis of *Lotus* in the United States, the case has been settled.[234] Therefore, the appellate court will not be given the opportunity to provide enlightenment.

Application of the Directive to the facts presented in the *Lotus* case most likely would yield results similar to the one reached by the Massachusetts district court.[235] In finding that the defendant's VP-Planner infringed the macro structure of Lotus 1-2-3, the district court noted that compatibility, which would allow users to transfer spreadsheets created in Lotus 1-2-3 to VP-Planner without the loss of functionality, could be achieved without the reproduction of the Lotus menu structure.[236] Evidence had been presented indicating that another commercially successful

[228] *See* Reichman, *supra* note 3, at 685-86.

[229] *See* Farrell, *supra* note 61, at 48.

[230] *Id.* at 48-49.

[231] 203 U.S.P.Q. (BNA) 124 (E.D. Mich. 1979).

[232] *Id.* at 128.

[233] *See* Reichman, *supra* note 60, at 957 n.95.

[234] Reed, *supra* note 12, at 50; Verdesca, *supra* note 210, at 1048 n.14.

[235] *See supra* text accompanying notes 216-19.

[236] Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 69 (D. Mass. 1990).

competitor had achieved compatibility simply by adding macro conversion capabilities.[237] Under the Directive, this evidence would indicate that reproduction was not indispensible to achieve interoperability, thus the article 6 decompilation exception would not excuse the actions of the defendant.[238] Furthermore, under the Directive, even the competitors that added the macro conversion capability may be deemed infringers if the decompilation exception of article 6 is not interpreted broadly to permit the development of competing products[239] because, in designing the macro conversion process, the competitors most likely decompiled the Lotus object code to determine the file formats and any other underlying interfaces. This narrow interpretation of the decompilation provision, however, would be unreasonable because one of the stated purposes of the Directive is to promote the ability of the EC software industry to compete with the dominant industry leaders.[240] Unless the decompilation exception permits the development of competing products, the goals of standardization and interoperability could never be reached.

As to the protection of the "look and feel" of the Lotus menu structure, whether the Directive will permit decompilation of a spreadsheet or any other software package that has become the industry standard is less certain. The issue of the menu structure or user interface differs from the issue of macro conversion because no compatibility concern is present. Decompilation simply would assist the second comer in the development and marketing of a competing product. Thus, decompilation under article 6 would not be permitted.[241] Nevertheless, a broad reading of the observe, study, and test exception found in article 5, paragraph 3, would permit reverse engineering, not rising to the level of decompilation,[242] to determine the underlying ideas and unprotectible elements of the lead vendor's spreadsheet package.[243] This broad reading is reasonable given the Commission's desire to promote the competitiveness of the EC software industry,[244] especially in the context of a spreadsheet mar-

[237] *Id.* Microsoft Excel Macro Translation Assistant was cited as a compatible competitor that had achieved compatibility with the Lotus spreadsheets without infringing. *Id.*

[238] *See* Final Directive, *supra* note 10, art. 6(1), 1991 O.J. (L 122) at 45.

[239] *See supra* notes 134-39 and accompanying text.

[240] *See supra* notes 33-34 and accompanying note 32.

[241] *See supra* notes 131-34 and accompanying text; Final Directive, *supra* note 10, art. 6(1), 1991 O.J. (L 122) at 45.

[242] *See supra* note 126 (discussing the distinction between noninfringing reverse engineering and decompilation).

[243] *See supra* part II.C.2.a.

[244] *See supra* notes 33-35 and accompanying text.

ket in which the lead vendor uses the threat of litigation to preserve market dominance.[245]

V. Conclusion

Adoption of the compromise Directive is a laudable step toward legitimizing the process of reverse engineering and promoting international standards for the protection of computer software. Nonetheless, one must not forget that the Directive was the result of a compromise between diametrically opposed lobbying groups.[246] Unlike the reverse engineering stance advocated by copyright scholars,[247] the decompilation exception found in the compromise Directive is severely limited by the requirement that decompilation be indispensible to interoperability. This interoperability limitation simply sweeps too broadly because it prohibits decompilation for the purpose of discovering underlying ideas. A broader right to reverse engineer, including the right of decompilation to discover unprotectible ideas, would have better promoted the EC software industry's desire to break into the international software market.[248]

Given this overbroad limitation on decompilation, EC tribunals should interpret the article 5 observe, study, and test exception broadly to permit noninfringing reverse engineering to discover unprotectible elements and ideas.[249] Furthermore, this interpretation properly reflects the compromise inherent in the Directive: the EC Council of Ministers adopted the interoperability limitation on decompilation, thus reflecting the concerns of SAGE,[250] meanwhile allowing the observe, study, and test exception—originally proposed by ECIS—to remain.[251]

Unfortunately, however, industry giants lobbied fiercely to achieve the short-term benefits afforded under a protectionist intellectual property scheme.[252] These industry giants, dominated by United States firms, fail to recognize that their objective should include steps to ensure competiveness in an international economy.[253] The United States and industry leaders like IBM and Lotus, fearful of Japanese competition, myopically

[245] *See supra* note 207.

[246] *See supra* part II.B.

[247] *See supra* notes 168-188, 192 and accompanying text.

[248] *But see* Weichselbaum, *supra* note 115, at 1065-68 (arguing that to permit decompilation, even when indispensible to interoperability, may be misguided).

[249] *See supra* part II.C.2.a.

[250] *See supra* nots 53-55, 71 and accompanying text.

[251] *See supra* notes 112-14 and accompanying text.

[252] *See supra* part II.B.; Jacobs, *supra* note 58, at 104.

[253] *See* Jacobs, *supra* note 58, at 104.

insist on an unlevel playing field in an attempt to preserve their current market dominance. This short-sightedness, however, injures not only the developing EC software industry, but someday may backfire and hinder current industry giants. Thus, contrary to the connotation of its name, the lobbying group SAGE blindly overlooks the wisdom behind a protective legal paradigm that permits legitimate reverse engineering and then handles any abuses at the infringement stage. For if and when another competitor, such as the Japanese software industry, develops a product that becomes the industry standard, who then will be begging for the economically sensible right of reverse engineering to better promote competition and innovation?

Linda G. Morrison