Pace Law Review

Volume 13
Issue 3 *Winter 1994*

Article 5

January 1994

Fair Use or Foul Play? The EC Directive on the Legal Protection of Computer Programs and Its Impact on Reverse Engineering

Marc A. Ehrlich

Recommended Citation
Marc A. Ehrlich, *Fair Use or Foul Play? The EC Directive on the Legal Protection of Computer Programs and Its Impact on Reverse Engineering*, 13 Pace L. Rev. 1003 (1994)
DOI: https://doi.org/10.58948/2331-3528.1420
Available at: https://digitalcommons.pace.edu/plr/vol13/iss3/5

Comment

Fair Use or Foul Play? The EC Directive on the Legal Protection of Computer Programs and Its Impact on Reverse Engineering

I. Introduction

In the face of a worldwide recession the software industry continues to show healthy growth.[1] The greatest percentage of this growth is taking place in Europe.[2] The European Economic Community ("EEC" or "EC") comprised of over 350 million European consumers[3] is, thus, a fertile market for commercial exploitation of computer software. In recognition of the proliferation of software into its member states, the EC has recently adopted the Council Directive[4] on the Legal Protection of Computer Programs ("Directive").[5] The Directive was drafted

1. *See, e.g.*, Steven Burke, *International Sales a Major Factor in Novell's 71% Earnings Increase*, PC WEEK, Mar. 9, 1992, at 157; Pierre A. Conseil, *Software More Promising Than Computer Services in Europe of the 1990's*, COMPUTERGRAM INT'L, June 16, 1992, at CGIO6160022.

2. Conseil, *supra* note 1, at CGI06160022 (projecting growth rates of approximately 20% through 1995).

3. Carl T. Hall, *The Pact Nobody Understands, North American Free Trade Agreement Puzzles Most*, S.F. CHRON., Sept. 7, 1993, at B1.

4. Directives are legally binding acts of the EC as to the results they seek to achieve. PENELOPE KENT, EUROPEAN COMMUNITY LAW -(THE M+E HANDBOOK SERIES) 24 (1992). They allow member states a measure of discretion as to the means of implementation, legislation or legislative action. They are the primary means that the EC uses to harmonize the states' laws. *See id.* at 25. The EC has four major sources of law: (1) the treaties creating the institution; (2) subsidiary treaties, called primary legislation; (3) secondary legislation, including regulations, directives, decisions, recommendations, and opinions; and (4) decisions of the European Court of Justice ("ECJ"). *Id.* at 23.

5. Council Directive 91/250 on the Legal Protection of Computer Programs, 1991 O.J. (L 122) 42 [hereinafter Directive].

with an eye towards the expanding European software market, with the understanding that computer programs are not protected in all of the member states, and that such protection where it exists, has different attributes.[6] The Directive is an ambitious piece of legislation. It attempts to impart a specificity to copyright protection for computer programs that is absent in its United States and Japanese counterparts.[7] Some commentators applaud these efforts.[8] Critics, however, while acknowledging the magnitude of the undertaking, fear that the Directive may further obfuscate this complex area of copyright law.[9] They worry that in its zeal to encourage competition and remove barriers to free trade, the EC may have ignored the historical underpinnings of copyright law, and the practical realities of the software industry.[10]

The legal protection of computer programs has been a troublesome issue since the infancy of the industry.[11] The United States, and most of the member states of the EC,[12] have chosen copyright law as the means to protect computer programs.[13] This has created problems, as copyright protection extends to the particular expression of an idea, and not the idea itself.[14] Application of copyright law to computer programs has raised questions regarding which elements of a computer pro-

---

6. *See id.*

7. *See* Alan K. Palmer & Thomas C. Vinje, *The EC Directive On The Legal Protection Of Computer Software: New Law Governing Software Development*, 2 DUKE J. COMP. & INT'L L. 65 (1992).

8. *See* James R. Warnot, Jr., *Software Copyright Protection In The European Community: Existing Law And An Analysis Of The Proposed Council Directive*, 6 SANTA CLARA COMPUTER & HIGH TECH. L.J. 355, 376 (1991) (inconsistencies with multinational conventions concerning copyright protection can be reconciled, and the Directive is a workable step toward providing consistent software protection in the EC).

9. *See, e.g.*, Palmer & Vinje, *supra* note 7, at 86-87; Mindy Weichselbaum, Comment, *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, 1061-68 (1990-1991).

10. *See* Palmer & Vinje, *supra* note 7, at 86-87.

11. *See* RAYMOND T. NIMMER, THE LAW OF COMPUTER TECHNOLOGY ¶ 1.01 (Supp. 1991).

12. See *infra* part III on the member states protection of computer programs.

13. *See, e.g.*, 17 U.S.C. §§ 101-117 (1988).

14. *See* PAUL GOLDSTEIN, COPYRIGHT, PATENT, TRADEMARK AND RELATED STATE DOCTRINES 558 (3d ed. 1990); *see also* H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667.

gram are ideas, and which are the expressive components of the underlying idea.[15] The extent to which the instructions that comprise the computer program are protected is of vital importance in determining the degree of reverse engineering that will be permitted under copyright law.[16] If one takes the position that a program's code is an expression, distinct from the program's underlying idea, then decompilation[17] cannot be permitted under copyright law.[18] Computer programs are especially vulnerable to piracy because they are expensive to develop and inexpensive to copy.[19] The Directive, while acknowledging this susceptibility, permits reverse engineering of computer programs with certain limitations.[20]

Section II of this Comment will examine how the Directive was adopted. Section III will examine the laws of some of the member states, and the steps that must be taken by these states to comply with the Directive. Section IV will compare the Directive with the Berne Convention, a longstanding multilateral copyright convention. In Section V, a comparison of the

---

15. *See, e.g.*, Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987). A computer program may be loosely defined as a detailed series of instructions that, when executed by a computer, bring about a desired result. *See* 17 U.S.C § 101 (1988). Computer programs are initially written in human-readable form (high-level programming languages) called source code, and then are converted through a program called a compiler to object code. NIMMER, *supra* note 11, at ¶ 1.03[2]. Object code is essentially a string of 1's and 0's that are read as on/off signals for the electronic components in the computer. *Id.* Object code is thus read by the computer, but would be indecipherable to the user of the program. *See* GOLDSTEIN, *supra* note 14, at 823.

16. *See* Weichselbaum, *supra* note 9, at 1035-39. Reverse engineering in the context of computer programs refers to the creation of a new program based on information obtained from the original. *Id.* at 1037. Reverse engineering of computer programs may be roughly divided into two categories: black box analysis and decompilation. Warnot, *supra* note 8, at 78-84. Black box analysis involves subjecting a program to certain input conditions, and monitoring its output in an attempt to better understand its function. *Id.* at 78-80. Decompilation on the other hand is a much more intrusive act, involving the conversion of the machine-readable object code back to source code. *Id.* at 80-84.

17. *See supra* note 16.

18. *See* 7 U.S.C. § 117 (1988).

19. Directive, *supra* note 5, at 42. "Whereas the development of computer programs requires the investment of considerable human, technical and financial resources . . . computer programs can be copied at a fraction of the cost needed to develop them independently." *Id.*

20. *Id.* art. 5-6, at 44-45. The Directive contains allowances for both types of reverse engineering. Black box analysis is permitted without restriction. Decompilation is permitted only in certain instances. *Id.*

United States copyright law, as it applies to computer software, will lend insight into the relative differences in the respective markets. Finally, Section VI will focus on the potential impact the Directive will have on the growing European software industry.

II. The Directive

In June of 1988, the Commission[21] issued a Green Paper on Copyright and the Challenges of Technology.[22] The Green Paper discussed current European copyright law, and other alternatives for protecting computer programs.[23] The Green Paper requested comments on a myriad of topics involving the legal protection of computer programs.[24]

After considering numerous commentaries, the Commission issued a proposal for a Directive ("Proposal").[25] The funda-

---

21. The legislative bodies of the EC are: the Commission, the European Parliament ("Parliament"), and the Council of Ministers ("Council"). *See* KENT, *supra* note 4, at 10-18. The Commission is a seventeen member body with representatives from each member state. It is divided into Directorates General ("DG") of varying size, which individually cover matters concerning the EC, such as external relations and competition. *Id.* at 16. The Commission initiates EC action by submitting proposals to the Council. *Id.* at 17. The Council is comprised of representatives from the member states who are responsible for voting on legislation proposed by the Commission. *Id.* at 14. The Parliament is comprised of 519 Members of European Parliament ("MEP") that are directly elected by the populations of all of the member states. *Id.* at 11. Their role in the passage of EC legislation is one of consultation with the Commission and Council. *Id.*

22. Palmer & Vinje, *supra* note 7, at 67. A green paper is a working document drafted by the Commission addressing and seeking commentary on a substantive issue of a proposed directive. *Id.*

23. *Id.*

24. *Id.* at 67-68. The Green Paper discussed, *inter alia*, whether protection should be granted via copyright law or some other method, whether protection should apply to programs fixed in any form, whether reproduction should not be permitted without the approval of the right-holder, whether adaptation by a licensee for personal use within the basic scope of the license should be permitted, and what the term of this protection should be. *See* Green Paper on Copyright and the Challenges of Technology: Copyright Issues Requiring Immediate Action, COM(88)172 final at 186-94.

25. *See* Palmer & Vinje, *supra* note 7, at 67-68; *see also* Commission Proposal for a Council Directive on the Legal Protection of Computer Programs, 1989 O.J. (C91) 4 [hereinafter Proposal]. A proposal is part of the EC's legislative procedure for the passage of a directive. *See* KENT, *supra* note 4, at 20. Under the cooperation procedure, the Commission drafts a proposal, which is sent to the Council for deliberation, and the Parliament for its first reading. *Id.* The Council then adopts a common position by a qualified majority of its members. Within three months,

mental thrust of the Proposal was that computer programs were to be protected under copyright law as literary works.[26] The Proposal detailed, *inter alia*, those who would benefit from the protection, acts requiring the authorization of the copyright holder, exceptions to these restrictions, and the terms of the protection.[27]

Article 4 of the Proposal enumerated certain restricted acts which required the authorization of the right-holder.[28] Under Article 4, the reproduction of any part of a program, whether done in the course of running, viewing, transmitting, or storing, was to be restricted; any adaptation of the program was restricted as well.[29] Notably, the Proposal was silent on reverse engineering. However, since decompilation requires loading the program into memory, it was presumably restricted under the Proposal.[30]

A. *The Reverse Engineering Debate*

1. *The ECIS Arguments*

The Proposal ignited a powder keg of debate, and set the stage for lobbying on a scale rarely seen before in the branches

---

upon a second reading, the Parliament will either approve or reject the Council opinion. *Id.* If the Parliament approves, the Council may adopt the act; if they reject, the Council may adopt the act only upon a unanimous vote. *Id.* If the Parliament amends the draft, the Commission may revise its proposal. *Id.* Within three months of this amendment, the Council must either adopt the amended proposal by qualified majority, adopt Parliamentary amendments that were not incorporated into the proposal, or otherwise amend the proposal by unanimous vote. *Id.* If the council fails to act, the Commission proposal lapses. *Id.*

26. *See* Proposal, *supra* note 25, at 6. The Proposal was drafted by the Directorate General for Internal Market and Industrial Affairs (DG-III) with help from the Director General for Competition (DG-IV) and for Telecommunications, Information, Industries, and Innovation (DG-XIII). *See* Palmer & Vinje, *supra* note 7, at 68. The Proposal was drafted so that its terms would coincide with the protection afforded under the Berne Convention and the Universal Copyright Convention. *See* Proposal, *supra* note 25, at 7. It rejected the notion of *sui generis* legislation in the field of computer programs because of the advantages that inure to conformance with these longstanding conventions. *Id.*; see *infra* part IV for a discussion of these multinational conventions.

27. Proposal, *supra* note 25, arts. 1-7, at 9-12.

28. *Id.* art. 4, at 10-11. Restricted acts included the reproduction, translation or adaptation of the code. *Id.*

29. *Id.*

30. *Id.*; *see also* Palmer & Vinje, *supra* note 7, at 70.

of the EC.[31] Small and mid-sized software companies, fearing that the implied restrictions on reverse engineering would place them at a competitive disadvantage compared to larger companies, banded together in an alliance called the European Committee for Interoperable Systems ("ECIS").[32] ECIS maintained that reverse engineering, including decompilation, was a legitimate means of competition used throughout the world.[33] ECIS claimed that such a prohibition of decompilation would interfere with the creation of interoperable programs.[34] Such interference, it claimed, would serve to insulate the larger software concerns from competition from small and medium sized European software companies.[35]

2. *The SAGE Arguments*

In response to these arguments, representatives from the larger software manufacturers formed a lobbying group called the Software Alliance Groups for Europe ("SAGE").[36] SAGE countered the arguments of ECIS by claiming that the term reverse engineering, as applied to decompilation, is misleading, as decompilation, by its nature, necessarily requires unauthorized copying, whereas, in other contexts, reverse engineering is ac-

---

31. *See* Palmer & Vinje, *supra* note 7, at 71-73.

32. *See EC: Violent Emotions Surge in Brussels over the Impending Changes to Software Copyright Laws*, COMPUTER WKLY., July 19, 1990, at 16 [hereinafter *Violent Emotions*]. Membership in ECIS included companies such as Bull and Olivetti, among others. *Id.*

33. *See Copyright: Debate over Scope of Computer Software Protection*, EUR. REP., Jan. 27, 1990, at 6. ECIS claimed that reverse analysis is generally used for the purposes of analysis and innovation, and that it is mainly used in the open-systems and systems integration sectors. *Id.* If companies could no longer rely on these techniques, ECIS asserted that U.S. monopolies in the proprietary systems sector, such as IBM and DEC, would be reinforced. *Id.*

34. *See Violent Emotions*, *supra* note 32, at 19. The term interoperable has been defined as the ability for two programs to work together, implying accurate knowledge of all hardware, and software interface protocols. *See* Andrew Johnson-Laird, *Reverse Engineering of Software: Separating Legal Mythology from Actual Technology*, 5 SOFTWARE L.J. 340 (1992). As interpreted by the ECIS advocates, this interoperability included the interaction or similar function of two programs. *See Violent Emotions*, *supra* note 32, at 19.

35. *See* Palmer & Vinje, *supra* note 7, at 68-71.

36. *See EC: IBM and DEC Turn Up Piracy Pressure to Get Amendments to Software Copyright Law Dropped*, COMPUTER WKLY., Jan. 4, 1990, at 72. SAGE membership included larger computer concerns such as IBM, DEC, Apple, and Microsoft. *See* Palmer & Vinje, *supra* note 7, at 70-71.

complished by a mere physical dismantling of the original protected item.[37] SAGE argued that it is in the financial interests of larger manufacturers to provide access to the relevant portions of their programs needed to design interoperable programs, and that licensing agreements between the manufacturer and these other parties could adequately control the communication of the information needed to create these programs.[38] Absent this limited access, SAGE argued, unrestricted decompilation would dissuade companies from investing resources into the development of software products that were vulnerable to piracy.[39] Finally, SAGE lobbyists asserted that the fears of monopolistic closed systems were misplaced because EC competition law will prevent dominant manufacturers from forming such monopolies.[40]

B. *The Amended Proposal and the Resultant Directive*

The impact of this lobbying created divided opinions within the three legislative branches of the EC.[41] The Legal Affairs

37. *See Violent Emotions, supra* note 32, at 17.

38. IBM Europe, Comments On The Debate Over "Reverse Engineering" of Software 8 (Oct. 1989) (unpublished comment on file with the *Pace Law Review*) [hereinafter Comments on the Debate]. When third party development of attaching programs is permitted, the original program expands its function, and is thus rendered more attractive to the potential consumer. *Id.* Additionally, refusal to provide such information is contrary to other EC competition legislation. Thus, some SAGE advocates hold that interoperability arguments are redundant. *Violent Emotions, supra* note 32, at 18.

39. *See Copyright: Battle Rages on Software Protection Directive*, MULTINAT'L SERVICE, Feb. 1990, at 11. They further argued that the notion of reverse engineering a literary work is nonsensical, and that the expression of a computer program is complete in its object code, as would be the expression of a novel in its text. Comments on the Debate, *supra* note 38, at 8. Another tenet of the SAGE argument was that authorizing decompilation will shorten the time it takes competitor's products to get to market. *Id.* The profit reaped during this curtailed "lead time" would thus be severely reduced, and the developer would be forced to compete against an adaptation of his own work, thereby further reducing the development incentive. *Id.*

40. *Violent Emotions, supra* note 32, at 18.

41. *See Intellectual Property: Commission Tables Amended Directive on Software Protection*, MULTINAT'L SERVICE, Oct. 1990, at 15; *see also* Palmer & Vinje, *supra* note 7, at 72. Directorate-General III, the original drafters, sided with the SAGE arguments for strong protection. *Id.* DG's IV and XIII seemed more amenable to the ECIS arguments. *Id.* Lobbying efforts in Brussels created a similar split in the Parliament. *Id.* at 73; see *supra* note 21 for discussion of the three legislative bodies of the EC.

Commission of the Parliament submitted an opinion to the Commission that included amendments favorable to reverse engineering.[42] The Commission responded in the fall of 1990 by drafting a modified proposal ("Amended Proposal").[43] The Amended Proposal included provisions for reverse engineering, creating a limited exception for copying, and adapting code for the design of interoperable programs.[44] Article 5 of the Amended Proposal allowed the user to perform black box analysis on a program.[45] Article 5a allowed for reproduction of the code and translation of its form (or decompilation), where necessary, to achieve the creation or maintenance of an independently created interoperable program.[46] The Amended Proposal permitted this act on the condition that the decompilation was used solely to achieve interoperability and was limited to those portions of the program needed to achieve interoperability.[47] It provided that the information was not to be given to others unless it related to the creation of the independent program, and that the resultant product was not to be substantially similar to the expression of the original program.[48] Finally, the Amended Proposal cautioned that these allowances could not be interpreted to permit acts which conflict with the right-holders normal exploitation of the program.[49]

Following the Commission's publication, a new debate arose regarding whether the exception for interoperability would apply only to independently created programs that would interface with or attach to the original program, or whether it would be extended to permit the use of decompilation to create programs that serve the same function as the original (i.e. com-

---

42. *See* Palmer & Vinje, *supra* note 7, at 75.

43. *Id.* at 76.

44. *See* Amended Proposal for a Council Directive on the Legal Protection of Computer Programs, 1990 O.J. (C 320) 22. The Amended Proposal defined interoperability as the functional interconnection and interaction of programs. *Id.* at 23.

45. *Id.* art. 5, at 27.

46. *Id.* art. 5a, at 28.

47. *Id.*

48. *Id.*

49. *Id.* art. 5a(3) at 28. Section 3 of Article 5a claims to bring the provision under the control of the Berne Convention for the Protection of Literary and Artistic Works. *Id.*; see also *infra* part IV for a discussion of the Berne Convention.

peting programs).[50] A proposal backed by the United Kingdom would have limited the act of decompilation to the creation of "connecting-products".[51] This proposal was defeated amidst heavy lobbying by both sides.[52] By year end, the Council had voted to adopt a uniform position ("Common Position") on the draft, which replaced the United Kingdom's proposal with a provision that would allow decompilation to achieve interoperability regardless of whether the new product connected to the original or not.[53] The Common Position was forwarded to the Parliament for a second reading, along with a communication from the Commission explaining how the opinion of the Legal Affairs Committee had been incorporated into the Directive.[54] The communication detailed the decompilation allowances of the Directive, but cautioned that it would not permit the reproduction of parts of the original program's code for uses other than achieving interoperability.[55] Parliament acquiesced in the Council's Common Position, and the Council finally adopted the Directive, with the full support of the Commission, on May 14, 1991.[56]

As adopted, the Directive recognizes computer programs as literary works subject to copyright protection under the Berne Convention.[57] No originality requirement is applied other than that the work is the author's own intellectual creation.[58] Under Article 4, the author is granted the exclusive right to authorize

---

50. *See* Karl Schneider, *EC: Storm Rages Over Britain's Proposal to Ban Reverse Engineering*, ELECTRONICS WKLY., Nov. 28, 1990, at 15.

51. *Id.*

52. *See* Palmer & Vinje, *supra* note 7, at 77.

53. *Id.*

54. Communication From the Commission to the European Parliament, EUR. PARL. DOC. (SEC 87 final-SYN 183) 5 (1991). The communication stated that decompilation is to be permitted to the extent necessary to ensure the interoperability of an independently created program. *Id.* "Such a program may connect to the program subject to decompilation. Alternatively *it may compete* with the decompiled program and in such cases will not normally connect to it." *Id.* (emphasis added).

55. *Id.*

56. *See* Directive, *supra* note 5, at 42.

57. *Id.*; see *infra* part IV for a discussion of the Berne Convention.

58. Directive, *supra* note 5, art. 1(3), 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." *Id.*

restricted acts such as reproduction, translation, adaptation, arrangement, or any other alteration of a computer program, as well as the distribution of the program.[59] Article 5 provides for exceptions to these restricted acts.[60] Thus, to the extent that the restrictions of Article 4 prevent the normal use of the computer program, they do not apply.[61] The making of back-up copies of the program is likewise permitted under Article 5.[62] Reverse engineering, including observing, testing, or studying the function of the program in order to determine the underlying principles thereof (i.e. black box analysis), may be conducted without the authorization of the right-holder.[63] However, under Article 6, decompilation is permitted only by the licensee or his agents where such reproduction of the code is indispensable to obtaining the information needed to create an independent interoperable program, and such information is not otherwise readily available.[64] In attempting to achieve compatibility with the Berne Convention, Article 6 states that the results of such decompilation may not be used to conflict with the normal exploitation of the program by the right-holder.[65] Finally, the Directive states that any contractual provision that disallows reverse engineering, as authorized under Articles 5 and 6, is null and void.[66]

Member states were instructed to adopt such provisions necessary to comply with the terms of the Directive by January 1, 1993.[67] The Directive is unclear on certain points, and this

---

59. *Id.* art. 4, at 44.

60. *Id.* art. 5, at 44.

61. *Id.* art. 5(1), at 44.

62. *Id.* art. 5(2), at 44.

63. *Id.* art. 5(3), at 45. This analysis is permissible if done while performing any of the acts of loading, displaying, running, transmitting or storing the program. *Id.*

64. *Id.* art. 6(1), at 45. Decompilation is to be limited to the portions of the program needed to obtain interoperability, and the results may neither be used or distributed to others for purposes other than the creation of interoperable programs. *Id.* art. 6(1), 6(2), at 45. Such resultant programs may not be substantially similar in expression to the original program. *Id.* art. 6(2)(c), at 45. Thus, although they may compete with each other and perform the same function, the expressive portions of the programs cannot be substantially similar.

65. *Id.* art. 6(3), at 45.

66. *Id.* art. 9(1), at 45.

67. *Id.* art. 10(1), at 46.

ambiguity has worried both ECIS and SAGE lobbyists.[68] First, the Directive ambiguously defines the term "interoperable".[69] Despite this, interoperability is the threshold criterion for determining whether the act of decompilation is to be sanctioned.[70] Second, the Directive states that decompilation is to be limited to those portions of the program needed to achieve interoperability.[71] Even if one were to ignore the ambiguity concerns, it is unclear how this provision is to be enforced. Practically speaking, it is nearly impossible for one to know which portion of the object code contains the "interoperable" information.[72] Finally, the Directive concludes that such an allowance of decompilation will not be read to permit unreasonable interference with the right-holder's normal exploitation of the computer program.[73] Yet, as stated above, the Commission has defined interoperability to include the creation of competing programs.[74] These two positions seem at odds with each other.

III. Laws of the Member States

The twelve member states of the EC provide varying degrees of protection for computer programs.[75] Some states such as Belgium, Ireland, Luxembourg, and Portugal have neither statutory provisions nor case law regarding the protection of computer programs.[76] Other states, such as the Netherlands, have stayed the enactment of national laws regarding the protection of computer programs pending the passage of the Directive.[77] The states that have developed national legislation

68. *See Software Argument Left Unresolved by European Commission on Software Copyright*, INFOMATICS, May 15, 1991, at 13.

69. *See* Directive, *supra* note 5, at 43. Interoperability may be defined as "the ability to exchange information and mutually to use the information which has been exchanged." *Id.*

70. *Id.* art. 6(1), at 45.

71. *Id.* art. 6(1)(c), at 45.

72. *See* Johnson-Laird, *supra* note 34, at 345.

73. Directive, *supra* note 5, art. 6(3), at 45.

74. *See* Palmer & Vinje, *supra* note 7, at 77.

75. Directive, *supra* note 5, at 42. Member states of the EC include Germany, France, Spain, Italy, U.K., Greece, the Netherlands, Portugal, Ireland, Belgium, Luxembourg, and Denmark; *See* KENT, *supra* note 4, at 6-8.

76. *See* Warnot, *supra* note 8, at 363.

77. *Id.* at 363-64.

regarding software protection were instructed to adapt this legislation to conform with the provisions of the Directive by January 1, 1993.[78] France, Germany, the United Kingdom and Spain have the most extensively developed legislation regarding the protection of computer programs, and thus, are facing the difficult task of harmonizing their software protection laws with the Directive.[79]

Inability to achieve conformity with the Directive may result in proceedings before the European Court of Justice ("ECJ").[80] In the event of a conflict between the Directive and a domestic law, the question of priority depends to some extent, on the member state's approach to incorporating international treaties into national law.[81] There are two main approaches: the monistic approach holds that the international obligation takes effect as soon as the treaty has been ratified;[82] the dualistic approach holds that an international obligation does not take effect until it has been incorporated into domestic statutes.[83] Additionally, the constitutions of the respective states control the determination of the supremacy of EC law.[84] Thus, those states that must endeavor to harmonize their domestic laws with the Directive or risk censure are confronted with a

78. Directive, *supra* note 5, art. 10(1), at 46.

79. *See* Warnot, *supra* note 8, at 363-68.

80. *See* KENT, *supra* note 4, at 30-31. Article 189 of the EEC Treaty holds that directives are binding the result to be achieved, but allows the states discretion as to how to achieve these results. *Id.* at 30. Conflict over the relationship between EC law and the laws of the member states as it relates to directives, is a vexing issue. The controversy stems from the extent to which directives are to be held directly effective (i.e. the extent to which they give rise to rights or obligations upon which individuals may rely before national courts). *Id.* at 31-32. While directives are unquestionably vertically directly effective (i.e. provisions may be enforced against states), debate continues regarding the extent to which the enforcement of a directive may fall on an individual (i.e. horizontal direct effect). *Id.*

81. *Id.* at 37.

82. *Id.* The Netherlands, for example, adopts this approach. *Id.*

83. *Id.* The U.K., Italy, and Germany follow this dualistic approach. *Id.*

84. *Id.* The ECJ has endeavored in several of its holdings to set forth harmonizing criteria, which would establish a new legal order in which the member states have limited their sovereignty within certain fields, in order to partake of its benefits. *Id.* at 39. Principles of this new order include the limitation of sovereign rights in areas covered by the EEC Treaty, and a restriction on national courts from applying domestic law in such areas of conflict. *Id.* at 39-41.

task whose magnitude depends on the conformity of their present domestic law with the terms of the Directive.[85]

A. *France*

In 1985, the French Senate adopted law no. 85-660, which amended the Copyright Act of 1957 to specifically include computer programs ("les logiciels") among the categories of copyrighted works.[86] The 1985 law does not put forth an originality requirement for copyright protection, but it is arguably covered under a general provision of the 1957 act requiring that the title of an intellectual work be protected where it is of an original character.[87] The 1985 law restricts the unauthorized copying of a computer program other than for the making of back-up copies.[88] However, it also provides that the author cannot restrict a licensee from adapting the program to meet its needs.[89] As worded, this limited allowance for adaptation does not contemplate decompilation.[90] In its decisions following the 1985 act, the French Supreme Court has held that a computer program must embody an intellectual contribution by the programmer to merit copyright protection.[91]

French courts have been reluctant to recognize the supremacy of EC law.[92] However, under the Directive, France will need to amend its domestic laws to allow users to perform reverse engineering in some situations, which will mean an expansion of the reproduction and adaptation provisions of the existing law.[93]

---

85. *See id.* at 37-41.

86. Copyright Act of 1957, Law No. 57-298, J.O. 4143, (Fr.) *as amended by* art. 1-V, Law No. 85-660, J.O. 7495, *translated in* U.N. EDUC., SCIENTIFIC AND CULTURAL ORG. ET AL., COPYRIGHT LAWS AND TREATIES OF THE WORLD (1992).

87. *See id.* art. 5.

88. CLIFFORD CHANCE, THE EUROPEAN SOFTWARE DIRECTIVE 20 (1991) (referring to Law No. 85-660 and its restrictions on reproduction).

89. *Id.*

90. *Id.*

91. *See, e.g.*, Warnot, *supra* note 8, at 365 (discussing implications of Judgment of Mar. 7, 1986 (Atari, Inc.), Cass. ass. plén., 1986 Bull. Civ. I, No. 1, at 7 (Fr.)).

92. KENT, *supra* note 4, at 46. Actually the Cour de Cassation (the highest court of appeal) has been more favorable to EC law than the Conseil d'Etat (the supreme administrative court). *See id.*

93. CLIFFORD CHANCE, *supra* note 88, at 20-21.

B. *Germany*

The German Copyright Act of 1965, like its French counterpart, was amended in 1985 to include computer programs as literary works subject to copyright protection.[94] Since the Act contains no explicit originality requirement for copyright protection of computer programs, the task of setting this standard has been the province of the German courts.[95] The decision of the Bundesgerichtshof (Supreme Court) in *Suedwestdeutsche Inkasso KG v. Bappart & Burker Computer GmBH*,[96] ("*Inkasso*") required that the threshold of originality to be met by a copyrightable program was that it represent a creative effort, and the skills needed to develop it exceeded those of an average programmer.[97] This standard made it very difficult to determine whether a work in development would be granted copyright protection.[98] Opinions following the *Inkasso* decision have softened this originality requirement to a demonstration of an intellectual effort.[99] The German Copyright Act, as amended, requires the permission of the author for any reproduction of the protected program, including the making of back-up copies.[100] The Act also requires the author's permission for the exploitation and publication of a protected work.[101] The situation regarding reverse engineering is not explicitly addressed. However, the statute may be read to require the right-holder's permission to reverse engineer.[102]

German courts, with the notable exception of the Federal Tax Court, have generally accepted the supremacy of EC law.[103]

---

94. Copyright Act of 1965, § 2(1) (F.R.G.), *translated in* BUSINESS TRANSACTIONS IN GERMANY app. 18-2 (Bernd Rüster ed., 1993) [hereinafter German Act].

95. *See* Warnot, *supra* note 8, at 364.

96. Judgment of May 9, 1985 (Suedwestdeutsche Inkasso KG v. Bappart Burker Computer GmBH), Entscheidungen des Bundesgerichtshofes in Zivilsachen, [BGHZ] (F.R.G.), *translated in* 17 INT'L REV. INDUS. PROP. & COPYRIGHT L. 681 (1986).

97. *Id.* at 687-88.

98. *See* Warnot, *supra* note 8, at 364.

99. *Id.* at 364-65.

100. German Act, *supra* note 94, §§ 15(1), 16.

101. *Id.* § 15.

102. CLIFFORD CHANCE, *supra* note 88, at 22-24.

103. *See* KENT, *supra* note 4, at 48. In one particular case the Tax court adopted a rule contrary to the earlier decision by the ECJ in the same case. *Id.* An appeal to the Bundesgerichtshof resulted in a reversal. *Id.*

In order to conform to the Directive, originality requirements of German copyright law need to be lowered, and explicit allowances for reverse engineering need to be added to the German Act.[104]

C. *The United Kingdom*

The United Kingdom has recently adopted the Copyright, Designs and Patents Act of 1988.[105] The Act includes computer programs as literary works to be protected by copyright.[106] Originality requirements are not explicitly defined within the Act. However, case law on point has set the originality standard at a low level.[107] The Act applies the author's control over copying to the whole or any substantial part of the work, regardless of whether copying occurs directly or indirectly.[108] The Act defines copying as reproducing the work in any material form, including storing the work in any medium by electronic means and making intermediate copies that are incidental to some other use of the work.[109] Thus, without the permission of the author, the normal use of a computer program, which entails some loading of the code into the memory of the computer, is restricted. The Act does not provide exceptions for the making of back-up copies, and would certainly disallow reverse engineering, including decompilation.[110] In addition to complete control over reproduction, the Act provides the author with extensive control over authorizing adaptations to the program.[111] The fair dealing provision of the Act provides that copying of

---

104. CLIFFORD CHANCE, *supra* note 88, at 23-24.

105. Copyright, Designs and Patents Act, 1988, ch. 48 (Eng.), *reprinted in* GERALD DWORKIN & RICHARD D. TAYLOR, BLACKSTONE'S GUIDE TO THE COPYRIGHT, DESIGNS & PATENTS ACT 1988, at 211 (1989) [hereinafter U.K. Act].

106. *Id.* § 3(1)(b), at 223.

107. *See, e.g.*, Interlego A.G. v. Tyco Indus. Inc., [1989] 1 App. Cas. 217 (P.C. 1988) (appeal taken from H.K.). The court held that copyright law will protect anything which is an independent product of the author, which involves the expenditure of some skill, labor, or experience, and which is not a slavish copy of another work, with no significant additional features. *See id.* at 256-59.

108. ROBERT MERKIN, COPYRIGHT, DESIGNS AND PATENTS: THE NEW LAW § 14.10-14.17 (1989).

109. U.K. Act, *supra* note 105, § 17(1), (6), at 229.

110. *See* MERKIN, *supra* note 108, § 14.15-14.16.

111. U.K. Act, *supra* note 105, § 21, at 230-31. The Act explains that "[i]n relation to a computer program a 'translation' includes a version of the program in which it is converted into or out of a computer language or code or into a different

literary works for the purposes of research or private study, criticism, or review, or reporting a current event does not constitute infringement, provided that sufficient acknowledgment of the work is included.[112] As such, it is unlikely that reverse engineering for the commercial exploitation of a computer program would be permitted under the fair dealing provision.

The courts of the United Kingdom have moved from reliance on national legislation over EC Law, to general acceptance of the supremacy of EC law, as evinced by both the Court of Appeal and the House of Lords.[113] A recent decision by the ECJ in a case referred to it by the Queen's Bench Divisional Court holds that national laws which challenge relevant EC law may be temporarily suspended, and interim relief granted under EC law.[114] The U.K. faces some challenges in harmonizing its domestic law with the terms of the Directive.[115] The Act would have to be drastically modified to permit the copying and translation necessary for decompilation.[116] Legislative amendments will be necessary to bring about this level of conformity.[117]

D. *Spain*

Spain has recently adopted a new law that labels computer programs as protected works and extends protection to source code and object code, as well as accessory documentation such as user manuals.[118] The Spanish law calls for absolute freedom between parties in agreeing to the conditions upon which rights

---

computer language or code, otherwise than incidentally in the course of running the program." *Id.* § 21(4), at 231.

112. *Id.* §§ 29-30, at 233-34.

113. *See* KENT, *supra* note 4, at 42-45. The House of Lords has applied a "rule of construction" approach to reconcile EC and UK legislation, loosely interpreting UK legislation so it will comport with relevant EC law. *Id.* at 43. If the UK legislation directly contravenes the EC law, the House of Lords can follow the domestic law; however, this scenario is rather unlikely. *Id.* at 44. Where the EC law is not directly effective, a compromise position is advocated, which involves interpreting the UK law 'purposively' to comply with the spirit and purpose of the relevant EC law. *Id.*; see *supra* note 80 and accompanying text for a discussion of direct effect.

114. *See* KENT, *supra* note 4, at 45.

115. CLIFFORD CHANCE, *supra* note 88, at 31-32.

116. *Id.*

117. *Id.*

118. Copyright Act of 1987, arts. 10(1)(i), 96(1), 96(2) (Sp.), *translated in* U.N. EDUC., SCIENTIFIC AND CULTURAL ORG. ET AL., COPYRIGHT LAWS AND TREATIES OF THE WORLD (1992).

to a computer program will be transferred.[119] Although reverse engineering is not mentioned in the legislation, the Spanish law closely follows the trends underlying the Directive regarding reproduction and adaptation.[120] Thus, although provisions regarding reverse engineering must be added to the law, the task of harmonizing Spanish law with the Directive should not prove as burdensome as that of its sister states.

The member states face a task of varying difficulty in attempting to harmonize their domestic laws with the Directive. Ironically, the states with the most developed software protection laws face the most arduous task in achieving conformance. This dichotomous result may have implications for the practicality of the Directive, as these states have developed their software laws based on years of experience with the software industry.

IV. Multinational Conventions

Desire to achieve uniformity among the rights conferred under copyright law was the impetus that lead to the drafting of multinational conventions on copyright.[121] Early bilateral treaties proved unsatisfactory in this regard as they created a patchwork of differing relationships between states.[122] Currently, several multinational copyright conventions are in force. The most prevalent of these is the Berne Convention for the Protection of Literary and Artistic Works ("Berne Convention" or "Convention").[123] Enacted in 1886, the Berne Convention

119. CLIFFORD CHANCE, *supra* note 88, at 27-29.

120. *Id.*

121. *See* STEPHEN M. STEWART, INTERNATIONAL COPYRIGHT AND NEIGHBOURING RIGHTS 33-35 (2d ed. 1989). Stewart explains this impetus by way of a 'home-made product tax' analogy. *Id.* at 35. If a work produced in country A were to be protected by copyright in country A but not in country B or C, so that it could be freely reproduced in those countries, it could be imported into country A where it would compete with products that have paid for the copyright permission. *Id.* As the imported products will not have had to pay for copyright permission, they will be cheaper and ultimately drive the home-made products out of the market. *Id.* The effect is as though a tariff had been imposed on domestic products, with no corresponding mark-up on imports. The greater the mobility of the goods the more serious the problem becomes. *Id.*

122. *Id.* at 36.

123. Berne Convention for the Protection of Literary and Artistic Works, June 26, 1948, S. TREATY DOC. NO. 27, 99th Cong., 2d Sess. (1990), 331 U.N.T.S. 217 [hereinafter Berne Convention]; *see also* STEWART, *supra* note 121, at 36-37.

promotes "national treatment" of copyright laws.[124] Thus, the right-holder is entitled to the copyright protection afforded such works in the country in which the work is being marketed, provided such laws comport with the parameters of the Convention.[125] An additional benefit of the Berne Convention, absent in other conventions, is that it automatically provides protection in other Berne states through membership in the Berne Union, without the attendant formalities of the particular marketing state.[126] Although the Berne Convention does not specifically list computer programs among the examples of copyrightable works, it is well accepted that computer programs are afforded protection under the Convention as literary works.[127]

Under the Berne Convention, protection is to be afforded to published as well as unpublished works.[128] Authors are granted the exclusive right to make and authorize translations of their works.[129] Authors are also granted exclusive control over the reproduction of their works with the exception of certain limited allowances for reproduction by the press.[130] Article 10 details a fair practice provision, allowing limited unauthorized reproductions of parts of a work, which have been made available to the

---

124. Warnot, *supra* note 8, at 361-62; *see also* STEWART, *supra* note 121, at 37.

125. STEWART, *supra* note 121, at 37-41. The success of the Berne Convention is due in large part to this use of national treatment. *Id.* at 36-37. Many of the Pan American conventions that were concurrent with the Berne Convention can attribute their failure to the adoption of the principle of lex fori, or the adoption of the laws of the country of the work's origin. *Id.* at 37. This impractical measure had courts in country A attempting to apply the laws of country B and C in copyright suits involving products created in those states. *Id.* Additionally, this scheme did not provide opportunities for nationals living in countries with weak copyright protection to experience the healthier protection of other states, and consequently lobby their own governments to improve the level of protection. *Id.* at 38.

126. Berne Convention, *supra* note 123, art. 4(2), S. TREATY DOC. NO. 27, *supra* note 123, at 3, 331 U.N.T.S. at 223.

127. *See* Warnot, *supra* note 8, at 362. The fact that computer programs are not included among the examples listed in the Berne Convention is understandable when one considers that the industry was in its infancy in 1971 when the Berne Convention was last amended. *Id.*

128. Berne Convention for the Protection of Literary and Artistic Works, June 26, 1948, art. 3(1)(a), *reprinted in* STEPHEN M. STEWART, INTERNATIONAL COPYRIGHT AND NEIGHBOURING RIGHTS 909, 910 (2d ed. 1989).

129. Berne Convention, *supra* note 123, art. 8, S. TREATY DOC. NO. 27, *supra* note 123, at 7, 331 U.N.T.S. at 229.

130. *Id.* art. 9, S. TREATY DOC. NO. 27, *supra* note 123, at 7, 331 U.N.T.S. at 229.

public for educational or newsworthy purposes.[131] The Berne Convention also grants control to authors over any adaptations, arrangements, and other alterations of the work.[132]

A. *Potential Conflicts with the Directive*

EC law is directly effective, and directly applicable.[133] International copyright conventions such as the Berne Convention, on the other hand, "merely oblige contracting states to bring their national legislation into conformity with convention law."[134] Where a contracting state fails to conform, the only remedy is an appeal by the injured state to the International Court of Justice ("ICJ").[135] Individual citizens of these contracting states have no remedy under the conventions on their own.[136]

Due to the Berne Convention's adherence to "national treatment", the effect of the Convention laws in a particular state is dependent upon the national laws of that state.[137] Convention provisions are incorporated into (and form part of) the law of the state, to the extent allowed under its national law.[138] This approach contrasts with EC law, which is not international law in the generally accepted sense, but is supranational (and thus *sui generis*) in nature as it endeavors to approximate the laws of the member states.[139]

Conflicts are inevitable between convention law and EC law. While convention laws are to be incorporated into the laws of the member states, EC laws attempt to harmonize the law

---

131. *Id.* art. 10, S. TREATY DOC. No. 27, *supra* note 123, at 7, 331 U.N.T.S. at 230-31.

132. *Id.* art. 12, S. TREATY DOC. No. 27, *supra* note 123, at 9, 331 U.N.T.S. at 233.

133. *See* KENT, *supra* note 4, at 27. Directly applicable laws confer rights and obligations on the governments (and in some cases the individuals) of a member state, which may be enforced in the national courts. *Id.*; see also *supra* note 80 and accompanying text for a discussion of the direct effect of EC law on the member states.

134. STEWART, *supra* note 121, at 561.

135. *Id.*

136. *Id.*

137. *Id.*

138. *Id.*

139. *Id.*

among the member states.[140] EC law cannot free the member states from obligations arising under international conventions, and cannot possibly account for all of the obligations of one or more member states under such conventions.[141] Conflicts between these two laws would be adjudicated before the ECJ, which has not yet faced this type of issue.[142] However, it is not within the jurisdiction of the ECJ to adjudicate points of Convention law.[143] Such conflicts could only be resolved by deciding whether or not a provision of Convention law should be recognized and applied within the EC.[144] The ECJ may refuse to apply a Convention law only if its provisions conflict with EC law.[145]

In the introduction and throughout the body of the Directive, the Council purports to adhere to, and conform with, the Berne Convention.[146] A close reading of the Directive, however, reveals several potential areas of conflict between the two laws. The Berne Convention grants the author control over any translation of his work,[147] whereas Article 5 of the Directive allows for unauthorized translations, and Article 6, insofar as decompilation requires translation, condones this unauthorized act as well.[148] Unauthorized reproductions are limited under the Berne Convention to uses that comport with fair practice (ie. uses that benefit the public such as news reporting, education, etc. . .).[149] The Directive permits reproduction as a step in the reverse engineering of a program.[150] As the motive behind this reproduction is often the commercial exploitation of a competing program, it would seem to conflict with the "public benefit" exceptions enumerated in the Berne Convention.

---

140. *Id.*

141. *Id.* at 561-62.

142. *Id.* at 562.

143. *Id.*

144. *Id.*

145. *Id.*

146. *See* Directive, *supra* note 5, at 43-45.

147. *See* Berne Convention, *supra* note 123, art. 8, S. TREATY DOC. NO. 27, *supra* note 123, at 7, 331 U.N.T.S. at 229.

148. *See* Directive, *supra* note 5, arts. 5-6, at 44-45.

149. *See* Berne Convention, *supra* note 123, arts. 10-10bis, S. TREATY DOC. NO. 27, *supra* note 123, at 7-8, 331 U.N.T.S. at 230-31.

150. *See* Directive, *supra* note 5, arts. 5-6 at 44-45.

Conflicts between the Berne Convention and the Directive present problems for the states. It remains to be seen whether conformity between these authorities can be achieved without resort to the courts. If this is not accomplished, the ECJ may refuse to recognize conflicting portions of the Berne Convention that cannot be reconciled with the Directive.

V. United States Law

A. *General Application of Copyright Law to Software*

1. *Legislation*

Due to its pioneering role in the software industry, the government of the United States was faced with the complexities inherent in drafting legislation to protect software before the governments of the member states.[151] In 1978, the Commission On New Technological Uses Of Copyrighted Works ("CONTU") finalized its report to Congress, making the United States the first nation that sought to explicitly include computer programs among the works to which copyright protection should apply.[152] This report resulted in the 1980 amendments to the United States Copyright Act of 1976 ("U.S. Act").[153] These amendments preceded the addition of computer programs to the copyright laws of the EC member states, and were a model from which they could draft their software copyright laws.

The U.S. Act gives authors the exclusive right to reproduce and distribute software.[154] This act also gives authors the right to create and authorize derivative works, as well as the right to authorize all translations and adaptations of their work.[155] The Directive, while granting the right to control adaptations and

151. *See* Palmer & Vinje, *supra* note 7, at 66.

152. *See generally* U.S. NAT'L COMM. ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT OF THE NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS (1979). The CONTU report recommended that 17 U.S.C. § 101 be amended to include computer programs in the definition of copyrightable works, that rightful owners of a program should be able to adapt the program to their uses, and that 17 U.S.C. § 117 should be modified to allow all copying that occurs incidental to the normal use of copyrighted programs. *Id.* at 1.

153. *See generally* 17 U.S.C. §§ 101-117 (1988).

154. 17 U.S.C. § 106 (1988) (detailing exclusive rights conferred to authors of copyrighted works).

155. 17 U.S.C. § 106(2) (1988). Derivative works are defined as those works that are based on "one or more preexisting works, such as a translation, . . . con-

translations, does not grant exclusive control of derivative works to the author.[156] The 1980 amendments to the U.S. Act create two narrow exceptions to the author's exclusive reproduction right.[157] The first exception provides that users are free to make back-up copies of computer programs and copies that are an essential step in the utilization of the program.[158] The second exception, known as the fair use exception, is similar to the fair practice exceptions in the Berne Convention as it exempts normally unauthorized acts such as reproduction, if the use is deemed fair.[159] The U.S. Act lists examples of fair uses as those for the purpose of criticism, commentary, newspaper reporting, or teaching.[160] It then provides a test to be applied on a case by case basis to determine if a particular use will be considered fair.[161] The fair use test is an equitable inquiry that balances the rights of authors against concerns such as the public benefit, in determining which reproductions should be exempted from copyright protection.[162] Section 107 of the U.S. Act lists factors to be considered in determining whether a particular use is fair.[163] These factors include:

> (1) the purpose and character of the use; including whether such use is of a commercial nature or is for non-profit educational purposes;
>
> (2) the nature of the copyrighted work;
>
> (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
>
> (4) the effect of the use upon the market for, or the value of the copyrighted work.[164]

---

densation, or any other form in which a work may be recast, transformed or adapted." *Id.* § 101.

156. *See Directive*, *supra* note 5, arts. 4-6, at 44-45. Programs created via decompilation are not controlled by the author and would, in many instances, constitute a derivative work. *Id.* art. 6, at 45.

157. *See* 17 U.S.C. §§ 107, 117 (1988).

158. *Id.* § 117.

159. *See id.* § 107; *see also* Berne Convention, *supra* note 123, art. 10, S. TREATY DOC. NO. 27, *supra* note 123, at 7, 331 U.N.T.S. at 229-31.

160. 17 U.S.C. § 107 (1988).

161. *Id.*

162. *Id.*

163. *Id.* § 107(1)-(4).

164. *Id.*

2. *Case Law*

The courts have used the standard of substantial similarity as a step in determining whether a work has been copied.[165] If there is no evidence that an alleged infringer had access to the original work, the similarities between the two works must be so striking as to preclude the possibility of independent derivation by the defendant.[166] To perform such a substantial similarity analysis, courts have adopted the "abstractions test", first espoused by Judge Learned Hand in the case of *Nichols v. Universal Pictures Corp.*,[167] in which two plays involving star-crossed lovers were scrutinized for their similarities.[168] Judge Hand explained that such a task involved the determination of the level of abstraction in expressing the idea, beyond which copyright protection will not extend:

> Upon any work a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may be perhaps no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas", to which, apart from their expression, his property is never extended.[169]

This concept is best explained via a simple example: Suppose we were analyzing two portraits. At the lowest level of abstraction, we could say that these two objects are very different (i.e. different canvas material, paint composition, brush strokes, etc.). As we proceed to a higher level of abstraction we may analyze the subject matter of the paintings, and once again may determine that they are different (i.e. they are portraits of different people). But as we progress toward higher levels of abstraction, (i.e. as more detail is excluded) it is obvious that differences in the two works will become less apparent (i.e. they are both portraits, both paintings, both works of visual art, etc.). The abstractions test begins at the lowest level of abstraction, and progresses upward, leaving more and more detail out

165. *See* See v. Durang, 771 F.2d 141, 142 (9th Cir. 1983).
166. *See* Arnstein v. Porter, 154 F.2d 464, 468 (2d Cir. 1946).
167. 45 F.2d 119 (2d Cir. 1930).
168. *Id.*
169. *Id.* at 121.

as it goes. The definitive evaluation of substantial similarity turns on the choice of the proper level of abstraction in determining "the line between expression and what is expressed."[170]

The application of the U.S. Act to computer software has been the subject of intense and continuing court battles.[171] The proper separation of idea from expression, a difficult task in any copyright action, is further complicated when the work is of a functional nature such as a computer program.[172]

Modern treatment of the idea expression distinction in functional works derives from the Supreme Court's holding in the case of *Baker v. Selden*.[173] In that case, the plaintiff had written a book describing a bookkeeping system and had included sample ledger sheets that would allow one to practice the described system.[174] The defendant later published a book containing ledger sheets that were substantially similar to those of the plaintiff.[175] In the copyright infringement action that followed, the Court stated that although Selden's description of the bookkeeping system was protected, the system itself was not.[176]

This holding was the foundation for what is now 17 U.S.C. § 102(b).[177] The application of § 102(b) to computer programs holds that while the creative expression adopted by the programmer is rightfully protected by copyright law, protection is not available for the processes or methods embodied within

---

170. *Id.* Judge Hand suggested that this is to be done on a case by case basis: "Nobody has been able to fix that boundary, and nobody ever can." *Id.*

171. *See, e.g.*, Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. denied*, 464 U.S. 1033 (1984).

172. Peter A. Wald et al., *Standards for Interoperability and the Copyright Protection of Computer Programs, in* INTELLECTUAL PROPERTY/ANTITRUST 701 (PLI Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series No. 339, 1992).

173. 101 U.S. 99 (1879).

174. *Id.* at 100.

175. *Id.*

176. *Id.* at 102. "Such a book may be explanatory either of old systems, or of an entirely new system; and, considered as a book, as the work of an author . . . [b]ut there is a clear distinction between the book, as such, and the art which it is intended to illustrate." *Id.*

177. Wald, *supra* note 172, at 709.

the program.[178] The protection of such processes and methods is the proper scope of the patent laws.[179]

This analysis reveals the difficulties encountered in determining what portion of a computer program is expressive and what is an idea. A substantial body of case law has developed concerning how far copyright protection should extend beyond literal computer code.[180] Protection of non-literal elements in computer programs has been sought for intangible concepts such as the "look and feel" of the program, or its output display, or the "structure, sequence and organization" of the program.[181]

In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*,[182] the United States Court of Appeals for the Third Circuit held that a program, although not a precise reproduction or translation, was an infringement because it was an exact copy of the structure of the original program.[183] The court reasoned that since derivative works and compilations were granted copyright protection under U.S. copyright law, the structure and organization of a literary work could be part of it's expression protected by copyright.[184] In *Whelan*, the defendant had created a personal computer-based program to manage a dental laboratory, by translating the plaintiff's mainframe-based "Dentalab" program.[185] Although the defendant had clearly derived his code from the plaintiff's program, there was no literal copying *per se*, due to the fact that the defendant's code was written in a different language.[186] In this case, the court extended copyright protection beyond the literal code to the non-literal elements of a program which included the "structure, se-

178. *Id.* at 710 (quoting H.R. REP. No. 1476, 94th Cong., 2d Sess. 57 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670).

179. 35 U.S.C. §§ 100-101 (1988). The unprotected method in *Baker v. Selden* was the plaintiff's "system of accounting". *See* Baker v. Selden, 101 U.S. 99, 104 (1879).

180. *See, e.g.*, Computer Assocs. Int'l, Inc., v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

181. *Whelan*, 797 F.2d at 1248.

182. 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

183. *See id.* at 1224-27.

184. *Id.* at 1239.

185. *Id.* at 1226.

186. *Id.*

quence, and organization" of the program.[187] The court defined the sole idea underlying the program as the efficient management of a dental laboratory, thus, any other aspect of the program was presumed to be expressive.[188]

The *Whelan* decision quickly prompted criticism. In *Computer Associates International, Inc. v. Altai, Inc.*,[189] it was argued that the *Whelan* decision was flawed in its assumption that only one idea underlies any computer program, and that once this separable idea is identified, everything else must be expression.[190] The *Computer Associates* court viewed the ultimate function or purpose of a computer program as the combined result of interacting subroutines.[191] Since each of these subroutines can individually be viewed as a program, each may be thought to have its own "idea".[192] Thus, *Whelan's* "one program-one idea" theory was found descriptively inadequate.[193] The court detailed a three-step test, based on the abstractions theory advanced by Judge Hand, which was to be used to determine whether the non-literal elements of two or more computer programs were substantially similar.[194] The three step test, which is widely accepted as the current law in software copyright actions, was labelled the "abstraction-filtration-comparison" test.[195] The first step, "abstraction", involves the same analysis described by Judge Hand in the *Nichols* case.[196] The court is to dissect the alleged infringer's code, isolating each level of abstraction, beginning with the code itself, and ending "with an articulation of the program's ultimate function."[197] The next step, "filtration", involves the examination of "the [program's] structural components at each level of abstraction to de-

187. *Id.* at 1248.

188. *Id.* at 1238-40. "Because there are a variety of program structures through which that idea can be expressed, the structure is not a necessary incident to that idea." *Id.* at 1240.

189. 982 F.2d 693 (2d Cir. 1992).

190. *Id.* at 705 (quoting 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03(f)(3) (1991)).

191. *Id.*

192. *Id.*

193. *Id.*

194. *Id.* at 706.

195. *Id.*

196. *Id.* at 706-07.

197. *Id.* at 707.

termine whether their particular inclusion . . . was dictated by considerations of efficiency . . . required by factors external to the program itself; or taken from the public domain and [are] hence unprotectable."[198] In the third and final step, "comparison", the court has filtered out all of the non-protected elements including ideas, elements dictated by efficiency or external factors, and elements taken from the public domain, and is left with a nucleus of protected expression.[199] It is this "golden nugget" of expression that the court must then compare to the original work in deciding whether the defendant copied any of this protected expression.[200]

The court acknowledged that this test was, at best, a guideline for determining copyright protection for non-literal program elements.[201] It questioned the efficacy of the Copyright Act to prevent public access to a program's source and object codes.[202] It observed that while copyright adequately protects traditional literary works, computer programs are more of a hybrid work, as they are at the same time writings and functional components in the larger process of computing.[203] Finally, the court suggested that the legislative resolution of an issue such as this may necessitate a CONTU II.[204]

B. *Copyright Law & Reverse Engineering*

The U.S. Act does not take a definitive stand on reverse engineering of copyrighted material, as it has no equivalent to either Article 5 or Article 6 of the Directive.[205] It has been argued that the author's control over derivative works precludes the allowance of reverse engineering.[206] Since there is no explicit legislation on point, this issue has been largely the province of the courts, which have differed in their resolution of the

198. *Id.*
199. *Id.* at 710.
200. *Id.*
201. *Id.* at 712.
202. *Id.*
203. *Id.*
204. *Id.*
205. *See* Weichselbaum, *supra* note 9, at 1053; *see also* 17 U.S.C. §§ 107, 117 (1988) (excluding a reverse engineering exception for the author's exclusive rights in copyrighted works); Directive, *supra* note 5, arts. 5-6, at 44-45.
206. Weichselbaum, *supra* note 9, at 1053-54.

question.[207] In early cases involving reverse engineering of computer software, the courts tended to avoid directly addressing the issue.[208]

In *Hubco Data Products Corp. v. Management Assistance, Inc.*, the United States District Court of Idaho enjoined the plaintiff from developing code that enhanced the defendant's product via a process that involved both creating a printout of the defendant's code and copying the defendant's microcode inside the computer.[209] The court's decision was premised on the fact that object code was subject to copyright protection under the U.S. Act, and thus, the activities of the plaintiff constituted a possible infringement.[210] In dicta, the court reasoned that if Hubco had deciphered the code through reverse engineering, it would still have constituted an infringement.[211]

1. *The* Sega *Standard*

Recently, in *Atari Games Corp. v. Nintendo of America*[212] and *Sega Enterprises, Ltd. v. Accolade, Inc.*,[213] two federal courts have been directly confronted with cases that have called for a determination of the legality of reverse engineering of computer programs.[214] These two cases presented similar factual backgrounds. Both cases involved a dispute between a video game console manufacturer (the plaintiffs), and a video game cartridge manufacturer (the defendants).[215] The dispute involved the plaintiffs' maintenance of a software "lock" mechanism in their game consoles that only permitted game

207. *See, e.g.*, Vault Corp. v. Quaid Software Ltd., 847 F.2d 255 (5th Cir. 1988); Sega Enters. Ltd. v. Accolade, Inc., 977 F2d 1510 (9th Cir. 1993).

208. *See* Vault Corp. v. Quaid Software, Ltd., 847 F.2d 255 (5th Cir. 1988); Hubco Data Prod. Corp. v. Management Assistance, Inc., 219 U.S.P.Q. (BNA) 450 (D. Idaho 1983) (holdings in both cases did not reach the compilation issue).

209. *Hubco Data Prods.*, 219 U.S.P.Q. (BNA) at 455-58.

210. *See Hubco Data Prods.*, 219 U.S.P.Q. (BNA) 450, 452 (1983). Hubco's "Nilsson Method II" increased the capability of MAI's operating system software by printing out the data structures of the MAI operating system, enabling Hubco to remove system "governors" inserted by MAI, and thereby increase the systems performance. *Id.*

211. *Id.*

212. 975 F.2d 832 (Fed. Cir. 1992).

213. 977 F.2d 1510 (9th Cir. 1992).

214. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1993); Atari Games Corp. v. Nintendo of Am., 975 F.2d 832 (Fed. Cir. 1992).

215. *Id.*

cartridges that contained the proper software "key" mechanism to function.[216] In this manner, the console manufacturers hoped to control access to their game consoles, thereby forcing game cartridge manufacturers to sign licensing arrangements to gain compatibility between their cartridges and the consoles.[217] Both console manufacturers also developed and sold their own game cartridges for use in their consoles, and thus, were active competitors with the defendants.[218] In both cases the defendants did not find the licensing arrangements to their satisfaction, and decided to develop their own software "key" mechanism.[219] As an intermediate step in creating their own compatible cartridges, both defendants decompiled the plaintiffs' object codes (which they obtained by purchasing the plaintiffs' game cartridges) to discern the function of the "key" mechanism.[220] Although the *Sega* court found for the defendant,[221] and the *Nintendo* court for the plaintiff,[222] the decision in the *Nintendo* case was a product of the defendant's commission of a fraud upon the Copyright Office, and did not turn on the resolution of the reverse engineering question.[223] Both courts found an allowance under the copyright law for the decompilation of object code, however, the *Sega* case contained a much more detailed explanation of the court's reasoning.[224]

In the *Sega* case, the court typified the question of the legality of decompilation of computer programs as one of first impression.[225] In this case, the defendant, Accolade, used a two-step approach to gain compatibility with the Sega console (called Genesis).[226] First, Accolade obtained copies of three of Sega's video cartridges, and a Genesis console, and wired the console to a decompiler, which generated printouts of the source

216. *Sega*, 977 F.2d at 1515; *Nintendo*, 975 F.2d at 836.

217. *See id.*

218. *Sega*, 977 F.2d at 1514; *Nintendo*, 975 F.2d at 835.

219. *Sega*, 977 F.2d at 1514-15; *Nintendo*, 975 F.2d at 836.

220. *Sega*, 977 F.2d at 1515; *Nintendo*, 975 F.2d at 836.

221. *Sega*, 977 F.2d at 1532.

222. *Nintendo*, 975 F.2d at 847.

223. *Id.*

224. *See Sega*, 977 F.2d at 1513 n.1 (explaining that the analysis and conclusions in the case are consistent with those in the *Nintendo* case).

225. *Id.* at 1513-14.

226. *Id.* at 1514.

code on the cartridges.[227] Accolade engineers then studied these printouts looking for "areas of commonality" among them.[228] Through such experimentation, the engineers at Accolade were able to create a development manual detailing the requirements for creating a Genesis-compatible game.[229] Next, Accolade engineers utilized this design manual to create Genesis compatible cartridges.[230]

In the meantime, Sega had licensed with another software company to use its patented Trademark Security System ("TMSS") software in the Genesis.[231] The TMSS software contained an initialization sequence that required the inclusion of the letters "S-E-G-A" at a particular memory location on the cartridge as a criterion for compatibility with the console.[232] Accolade discovered that its cartridges did not work on these new Genesis consoles and sought to reestablish compatibility.[233] Eventually, Accolade learned of the "S-E-G-A" requirement, and added this requirement to its design manual, and, shortly thereafter, Accolade once again commenced releasing Genesis-compatible games.[234]

Sega filed suit in the district court claiming *inter alia*, that Accolade infringed its copyright on its game cartridges.[235] In granting Sega's motion for a preliminary injunction, the district court rejected Accolade's argument that intermediate copying of copyrighted code does not constitute infringement.[236] Accolade's appeal from that injunction was what prompted the circuit court's consideration.[237]

Accolade presented four arguments to support its contention that its decompilation did not amount to a copyright in-

227. *Id.* at 1514-15.

228. *Id.* at 1515.

229. *Id.* Accolade claimed that this manual contained only "functional descriptions" of Sega's code and did not contain the literal code. *Id.*

230. *Id.* Thus, Accolade maintained that it did not copy Sega's programs but instead relied on their own development manual. *Id.*

231. *Id.*

232. *Id.*

233. *Id.*

234. *Id.*

235. *See id.* at 1516.

236. *Id.*

237. *Id.*

fringement.[238] First, it argued that intermediate copying does not infringe on the author's exclusive rights, "unless the end product . . . is substantially similar to the copyrighted work."[239] Second, it maintained that decompilation is permitted under 17 U.S.C. § 102(b), if it is preformed to gain an "understanding of the ideas and functional concepts embodied in the code."[240] Third, it claimed that decompilation is allowed under 17 U.S.C. § 117, which allows a lawful owner of a copy of a computer program to load it into the computer's memory.[241] Finally, it contended "that disassembly of object code to gain an understanding of the ideas and functional concepts embodied in the code is a fair use that is privileged by section 107 of the Act."[242]

The court dispensed with Accolade's first three arguments as being at odds with the U.S. Act and relevant case law.[243] It found that intermediate copying was a violation of the author's exclusive rights under 17 U.S.C. § 106, regardless of whether the end product of the copying also infringed those rights.[244] The court conceded that there has been some scholarly authority for Accolade's assertion that object code should only receive "thin" copyright protection, as it is not readily human-readable, and is thus unlike other literary works for which the protection applies.[245] However, the court, disagreed with this contention, pointing to 17 U.S.C. § 102, which includes protection for works "which can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."[246] The court dispensed with Accolade's § 117 defense by opining

238. *Id.*

239. *Id.*

240. *Id.*

241. *Id.*

242. *Id.* at 1518.

243. *Id.* "Neither the language of the Act nor the law of this circuit supports Accolade's first three arguments." *Id.*

244. *Id.* The court found that although the "allegedly infringing copy of a protected work may itself be only an inchoate representation of some final product to be marketed commercially [this] does not in itself negate the possibility of infringement." *Id.* (quoting Walker v. University Books, 602 F.2d 859, 864 (9th Cir. 1979)).

245. *Id.* at 1519. An example of this authority can be found in Brief *Amicus Curiae* of Eleven Copyright Law Professors, Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992) (No. 92-15655), *reprinted in* 33 JURIMETRICS J. 147 (1992).

246. *Sega*, 977 F.2d at 1519-20 (quoting 17 U.S.C. § 102(a) (1988)).

that Accolade's use was beyond the intended scope of the § 117 allowance.[247]

The court finally addressed Accolade's contention that its decompilation was a fair use permitted by § 107.[248] Accolade argued that decompilation was the only means available in this situation to gain access to the unprotected elements in the program, and that it had a legitimate interest in gaining such access (i.e. the creation of Genesis-compatible cartridges).[249] The court agreed with this argument and created a decompilation exception to copyright protection of computer programs under the fair use doctrine.[250] The exception applies where there is a legitimate reason for examining the unprotected elements of the code and decompilation is the only available means of gaining such access to the code.[251]

The court arrived at this decision by weighing the four statutory factors delineating fair use under § 107.[252] With respect to the first factor, the nature of the use, the court dispensed with Sega's assertion that since the copying was done to produce a competing product, it was a commercial use, not entitled to fair use protection.[253] The court found that the use was an intermediate step in the production of a commercial work, thus the commercial exploitation was indirect.[254] It decided that although Accolade's ultimate purpose was the creation of Genesis-compatible games, its direct purpose, and thus its direct use of the material, was merely to understand the functional requirements for compatibility.[255] The court concluded that Accolade's use of Sega's code was for a legitimate purpose consistent with the fair use requirements and held that the presumption of unfairness incident to commercial use had been overcome.[256]

---

247. *Id.* at 1520.

248. *Id.* at 1520-28.

249. *Id.* at 1520.

250. *Id.* at 1527-28.

251. *Id.*

252. *Id.* at 1521-27.

253. *Id.* at 1522. The court characterized Sega's analysis as overly simplistic in that it ignored other aspects of the "purpose and character of the use." *Id.*

254. *Id.*

255. *Id.*

256. *Id.* at 1522-23.

The second statutory factor, the nature of the copyrighted work, prompted the court to discuss the "hybrid nature" of computer programs as literary and utilitarian works.[257] The court acknowledged the fact that object code is unlike traditional literary works as it is not directly human-readable.[258] The court further opined that, apart from decompilation, no other viable alternative existed for gaining access to the unprotected elements in Sega's code.[259] It pointed out that in such circumstances, the determination that decompilation was a *per se* unfair use, would be tantamount to granting a monopoly over the ideas embodied within the code, ideas that were statutorily denied copyright protection.[260] Thus, upon finding that Sega's code contained unprotected elements to which access could not be had without copying, it was afforded a lower level of protection than traditional literary works.[261] The court concluded that the nature of the work factor weighed in favor of Accolade's use.[262]

The third factor was the amount copied in relation to the work as a whole.[263] Here the court found that Accolade's decompilation necessarily involved the copying of the entirety of Sega's code.[264] Thus, this factor weighed in Sega's favor.[265] The court, however, asserted that this factor, by itself, was not determinative of the fair use question.[266] The court explained that in cases where the use was not directly commercial, this factor was to be accorded very little weight.[267]

---

257. *Id.* at 1524-25.

258. *Id.* at 1525. The court dispensed with Sega's contention that trained software engineers can, in fact, read object code, but that it is a tedious task. "Because even a trained programmer cannot possibly remember the millions of zeros and ones that make up a program . . . he must make a written or computerized copy of the disassembled code in order to keep track of his work." Thus, in the translation of object to source code copies must be made. *Id.*

259. *Id.* at 1526.

260. *Id.*

261. *Id.*

262. *Id.*

263. *Id.*

264. *Id.*

265. *Id.*

266. *Id.*

267. *Id.* at 1526-27.

The court then looked at the fourth statutory factor, the "effect on the potential market for the copyrighted work."[268] The court found that Accolade did not intend to preempt Sega from releasing a particular game into the market.[269] Rather, the court found that Accolade only sought to become a "legitimate competitor in the field of Genesis-compatible video games."[270] The court held that Accolade's entrance into the market of Genesis-compatible cartridge producers had an indirect effect on Sega's video game market.[271] There was no reason to believe, for example, that a purchaser of Accolade's "Ishido" cartridge would be dissuaded from buying Sega's "Altered Beast" cartridge.[272] Thus, the court found that the fourth statutory factor weighed in favor of Accolade.[273]

Accordingly, the court found that as the first, second, and fourth statutory factors weighed in favor of Accolade, it was likely to prevail on its fair use defense, and the preliminary injunction was unwarranted.[274] This holding, which must be considered the current law of the United States regarding decompilation, does not square with the traditional context of copyright law, at least to the extent that it permits the "wholesale copying" of the plaintiff's work for a commercial purpose.[275] The court explains this apparent discrepancy by pointing out that this is "a complex and relatively unexplored area of copyright law", and that its decision is in keeping with the policy considerations underlying the copyright law.[276]

The court dismissed an argument advanced by the plaintiff contending that much time and money went into the preparation of the Genesis system and its game cartridges, and that the court's holding would allow Accolade to be a "free rider" on the plaintiff's laborious product development.[277] The court pointed to the holding in *Feist Publications, Inc. v. Rural Telephone Ser-*

268. *Id.* at 1523.
269. *Id.*
270. *Id.*
271. *Id.*
272. *Id.*
273. *Id.* at 1524.
274. *Id.* at 1527.
275. *Id.*
276. *Id.*
277. *Id.*

vice Co.[278] which stated that "sweat of the brow" is not protected under the U.S. Act.[279] However, the work involved when preparing a phone book, as was the case in *Feist*, is arguably less creative than the development of computer software. If the copyright laws will not protect Sega's effort, then it may be argued that a fundamental purpose thereof, that of the promotion of the publication of ideas, is being frustrated. This decision may be construed as discouraging investments of time and money into the production of innovative software. In light of the importance of the industry and its rapid growth, this factor is deserving of serious consideration.

It is interesting to note that while both the Directive and the *Sega* decision would permit decompilation of object code under certain circumstances, neither is able to articulate with any degree of certainty what these circumstances are. Software developers are not given clear guidance on when their works are to be protected, or when their acts will constitute copying. Attorneys must grapple with these vague standards in attempting to advise their clients, and the courts, with such minimal guidance, are likely to arrive at inconsistent holdings, further complicating an area of law that is already wanting in clarity.

VI. Analysis: The Software Industry in the Wake of the Directive

The Directive was the EC's attempt to find a middle ground between the conflicting interests within the software industry.[280] It was hoped that a limited exception to the author's exclusive rights to allow for interoperability would balance the larger manufacturers' concerns about piracy with the smaller developers' fears over the bolstering of monopolistic proprietary systems.[281] As drafted, the Directive is certainly the most ambitious legislation to date in the area of software protection, and the first that has directly addressed the decompilation issue. The impact that the Directive will have on the industry remains to be seen. However, it is inevitable that the Directive will cre-

278. 499 U.S. 340 (1991).

279. *Sega*, 977 F.2d at 1527 (citing *Feist Publications*, 499 U.S. at 351-61).

280. See *supra* part II for a discussion of the reverse engineering exception, and the debate resulting therefrom.

281. *Id.*

ate problems as developers seek to enforce their varying interpretations on one another.

One point of contention already being debated is the provision under Article 6 stating that decompilation is not to be sanctioned unless it is indispensable to achieving interoperability.[282] A further provision therein states that the information needed to achieve interoperability must not have been readily available.[283] From this reading, one may assume that if information is required to achieve interoperability, a request may be directed to the manufacturer. The manufacturer will be unlikely to refuse such a request. In the first place, most manufacturers publish extensive system specifications with their products that contain much of the interface information that would likely be requested.[284] Additionally, the refusal of such a request, under the Directive, may trigger the user's decompilation rights.[285] Most software license agreements provide that information will be made available at some cost to the requester.[286] The Directive, however, provides no guidance on how this arrangement is to be managed. For instance, small developers may find the fee required to obtain the interface information needed for their interoperable design usurious. In such situations, will developers be able to decompile the licensed program, claiming that the requisite information was not readily available? This issue may necessitate the adoption of clarifying legislation in the area of software licensing.

Additionally, questions arise as to how much information the developer must provide.[287] Often there are many possible interfaces to a software application.[288] Must the developer provide all of these? Who will resolve disputes of this nature? The Directive is unclear on the degree of availability that is required to preclude the decompilation right from arising in the user.[289] Thus, in response to the inevitable disputes that will arise over

282. *See* Directive, *supra* note 5, art. 6(1), at 45.

283. *See* Directive, *supra* note 5, art. 6(1)(b), at 45.

284. *See Violent Emotions*, *supra* note 32, at 19.

285. *See* Hillary Pearson et al., *Commercial Implications of the European Software Copyright Directive*, THE COMPUTER LAW., Nov. 1991, at 17.

286. *Id.*

287. *Id.*

288. *Id.*

289. *See id.*

this issue, the states will have to further adapt their copyright legislation.

Another issue that will undoubtedly require further legislation is the standard by which the act of decompilation will be judged.[290] Users will need to seek technical and legal counsel before deciding to decompile a program in order to understand the extent of decompilation, if any, that is legally permissible on a particular program.[291] This level of uncertainty is a function of the open-ended definition for interoperability adopted in the Directive, and will undoubtedly add significant time and expense to software development cycles. As this lack of clarity becomes an issue in the industry, it will fall to the member states of the EC to add precision to the interoperability exception.

Some commentators claim that fears over the widespread use of decompilation as a vehicle for pirating software are baseless.[292] They point to the expense in technical skill, time and money that is required to perform decompilation, and the effort involved in trying to understand the tens (or hundreds) of thousands of lines of source code that typically result from decompilation.[293] This argument, however, is weakened considerably by the recent development of computer aided software engineering ("CASE") tools, which permit the decompiled source code to be quickly understood and analyzed, by graphically revealing the underlying structure of the decompiled program.[294] Thus, engineers will be able to quickly understand undocumented source code, and identify major components and data structures, facilitating the reverse engineering process.[295] This new segment of the software industry was born of the necessity to maintain and adapt existing software applications to the

290. *Id.* at 21.

291. *Id.*

292. *See, e.g.*, Johnson-Laird, *supra* note 34, at 347-48.

293. *Id.*

294. *See, e.g., FORTRAN: CADRE Technologies Intros TEAMWORK/FORTRAN REV for Reverse Engineering of Fortran Programs - Runs on SUN Today, IBM, HP & DEC Q4/91*, EDGE, July 22, 1991, *available in* LEXIS, News Library, NWLTRS File. Most of these tools currently require that the code has already been converted to source code. They then facilitate quick understanding of the source code. There are also tools available for converting object code to something like source code (pseudo-code), but these are dependent on the processor used to compile the original code. *Id.*

295. *Id.*

changing needs of the business environment; however, it is equally useful for those seeking to create a competing product.[296] Thus, when one considers the fact that in 1989 estimated profits lost to pirating operations in Germany, France, the United Kingdom and Spain totalled well over three billion dollars,[297] together with the emergence of a new industry that will facilitate the efficient reverse analysis of software, it is easy to see why concern over a decompilation exception is justified.

The ambiguity of the Directive has many industry experts concerned about the great potential it has to propagate litigation.[298] It has been typified as "a user's nightmare and a lawyer's dream."[299] Many feel that the opportunity for litigation under the Directive will place small manufacturers at the mercy of larger companies which can afford the costs of protracted law suits.[300] If this prediction holds true, many of the ambiguities within the Directive may ultimately be resolved in favor of the larger manufacturing concerns.

While it remains to be seen how the various ambiguities within the Directive will be resolved, it is a certainty that the early stages of its implementation will be fraught with litigation.[301] It is only through the respective state governments' interpretations of the Directive that it will provide the specificity needed to sustain industry growth. It is highly likely that the states will turn to the recent holdings of the United States courts in defining the legal parameters of decompilation. However, as the *Sega* holding indicates, United States case law has yet to lend sufficient clarity to this area of the law.[302]

---

296. *Id.*

297. *See* Mike Lewis, *What Software Copyright Laws Allow; European Community's Directive on Software Copyright International Report Related to the High Cost of Software Piracy in Europe*, DATA BASED ADVISOR, Mar. 1992, at 130.

298. *See EC Directive Angers Software Users; European Community Copyright*, BUS. & COMPUTER PUBLICATIONS, June 5, 1991, at 17.

299. *Id.*

300. *See id.*

301. *Id.*

302. See *supra* part V for a discussion on U.S. copyright law and reverse engineering.

VII. Conclusion

The European Council Directive on the Legal Protection of Computer Programs, which took effect on January 1, 1993, is the most ambitious legislation to date in the area of software protection. The purpose of this legislation was to harmonize the laws of the member states with regard to software protection and to strike a balance between the competing needs of large and small developers, including software users. At the same time, there was an attempt to add specificity to copyright protection of software that is lacking in national legislation, without exceeding the historical copyright protection of the Berne Convention. Such an undertaking is both necessary and laudable, as this industry promises to continue its healthy growth and should form a cornerstone of the European Community's economy.

The member states face challenges in attempting to conform their laws to the Directive. There is no state whose laws currently permit reverse engineering of computer programs. Moreover, the states' respective provisions regarding reproduction, translation and adaptation of software will need to be modified, if conformity is to be achieved. Additionally, as signatories of the Berne Convention, states must either face the task of interpreting portions of the Directive that are seemingly at odds with provisions of the Convention, or seek a ruling from the ECJ on whether portions of the Convention are to remain valid under EC law. Finally, it will fall upon the states to lend clarity to the Directive's ambiguous treatment of the definitions and limitations of the decompilation exception.

There is an inevitable tension that exists when drafting new legislation. It is desirable to lend sufficient detail to the law to ensure its utility, while avoiding the creation of iron-clad rules with no room for growth. This is especially true when the legislation involves a rapidly evolving area of technology such as computer software. Thus, the drafters of the Directive faced the challenges of providing protection for software developers without ignoring the changing nature of the industry.

The Directive creates exceptions to traditional copyright law tailored specifically to the protection of computer programs. However, it provides little detail on the extent or implementation of these exceptions. Additionally, it ignores many of the

practical realities of the current market. This imprecision will undoubtedly result in much needless litigation.

The Directive will not serve to enhance the growth of the software industry. Although the EC solicited extensive input from industry experts in drafting the Directive, it can at best be labelled a first step toward definitive software protection legislation. Until its imprecision and inconsistencies are removed, the Directive cannot accomplish the goal of uniform software protection within the EC, and is likely to create more confusion than existed prior to its passage. For now, the future of European software protection lies with the governments of the member states.

*Marc A. Ehrlich**

* This Comment is dedicated to my wife. I would also like to thank my family for their support.