Comparing U.S. and EC Copyright Protection for Computer Programs: Are They More Different Than They Seem

Pamela Samuelson

Follow this and additional works at: https://scholarship.law.berkeley.edu/facpubs

Part of the Intellectual Property Law Commons

Recommended Citation

This Article is brought to you for free and open access by Berkeley Law Scholarship Repository. It has been accepted for inclusion in Faculty Scholarship by an authorized administrator of Berkeley Law Scholarship Repository. For more information, please contact jcera@law.berkeley.edu.
Comparing U.S. and EC Copyright Protection for Computer Programs: Are They More Different Than They Seem?

Pamela Samuelson*

In May of 1991 the Council of the European Communities adopted a Directive on the Legal Protection of Computer Programs.1 This Directive required member states of the European Community to enact legislation to protect computer programs as literary works under copyright law.2 Also required is enactment of some special rules applicable to computer programs but not to other literary works.3 By adopting the Directive, the Council aimed not only to harmonize the laws of member states of the EC, but also, in part, to bring the law of the EC into conformity with the law of the United States as to the legal protection of computer programs.4 Now that the member states of the EC are

---


3 See Directive, supra note 1, arts. 5-6. These provisions are discussed infra parts B and C.

4 The principal objective of the Directive was, of course, to harmonize the laws of the member states of the EC about legal protection for computer programs. Directive, supra note 1, Recitals. For an overview of the pre-Directive state of European law regarding software protection, see, e.g., Hugh C. Hansen, Copyright Protection For Computer Programs in the EEC and Europe, in...
implementing the Directive, a question ripe for the asking is to what extent has the intended conformity between U.S. and EC law actually been achieved.

This article will suggest that although the EC Directive has brought the copyright law of the United States and of the EC concerning the protection of computer programs into harmony in a number of important respects, the Directive may have brought about less harmony than might be apparent at first glance.

There are three sources of potential disharmony between U.S. and EC copyright law as applied to computer programs. First, there are some commercially important fine details of the Directive and U.S. law that are significantly different. Among these details are rules regulating license contract terms.

Second, the law of the EC is largely fixed by the text of the Directive, whereas U.S. law is evolving through its normal, if sometimes tortuous, common law process. While some U.S. case law developments have brought U.S. law more in harmony with the Directive, in other respects, U.S. case law has evolved in different directions than may have been expected at the time the Directive was being considered, particularly as to the scope of copyright protection for internal design elements of programs.

Third, although the Directive is extremely detailed about such issues as the copyright status of decompilation and elements of programs necessary for interoperability, it is silent about whether copyright protection is available for some valuable aspects of software, such as user interfaces and program behavior. Unless EC courts decide to engage in

---

**Global Intellectual Property Series: Practical Strategies—Trademark and Copyright** (PLI Pats., Copyrights, Trademarks and Literary Prop. Course Handbook Series No. 318, 1991). Although harmonization of laws within the Community was the principal concern of EC intellectual property policy makers, they were very much aware of the U.S. decision to adopt copyright as a form of legal protection for computer programs and of the considerable interest U.S. trade negotiators and software developers had in the EC's decision about the use of copyright to protect programs. Furthermore, since trade in software products was likely to be international, they were very much aware of the desirability of relatively harmonized laws about protecting programs as between the EC and the United States.

5. Article 10.1 of the Directive required that member states implement the Directive in their national laws by January 1, 1993. Very few countries met this deadline, however, and even today, not all have done so. See Thomas C. Vinje, *Recent Developments in European Intellectual Property Law: How Will They Affect You and When?*, 13 J.L. & COMM. 301 (1994). This delay does not seem to be due to any national resistance to enactment of the Directive, but rather to other causes. In Belgium, for example, delay in enacting legislation implementing the Directive has occurred because Belgium is in the process of doing a substantial update of its copyright law, a process which, as U.S. policy makers know, can take a very long time.
common law decision making, these aspects of software may eventually be ruled unprotectable by courts applying EC copyright law. U.S. courts, in contrast, have construed U.S. copyright law as extending protection to at least some external aspects of computer programs.

A. PROTECTION FOR PROGRAMS AS LITERARY WORKS

The most obvious—and, of course, a highly important—similarity between U.S. and EC law concerning legal protection for computer programs is that both have now adopted copyright as a form of legal protection available to authors of computer programs. And both laws consider programs to be "literary works."\(^6\) Authors of such works are protected by copyright law from the moment of the first fixation of the work in a tangible medium as long as the programs satisfy the relatively minimal originality standard of copyright law.\(^7\) Both laws aim to protect the "expression" in programs, but not "ideas" and "principles" underlying them.\(^8\) Protection by copyright law under the Directive and U.S. law lasts for the life of the program's author plus fifty years.\(^9\)

---

6. Directive, supra note 1, art. 1.1. Czarnota and Hart, who were active participants in the drafting process of the Directive, state that programs "are not to be treated as a separate category of works, or assimilated in some way to literary works without, in fact, being totally admitted to that category." CZARNOTA & HART, supra note 2, at 30 (emphasis in the original). One goal of this approach was to get programs to be treated as literary works within the meaning of the Berne Convention so that the benefits of international protection under the Convention could be met. Id. at 31.

7. The Directive defines the term "originality" as meaning only that the program was the work of the author's own intellectual effort. Directive, supra note 1, art. 1.3. Prior to adoption of the Directive, the German Supreme Court had required a very high level of originality in order for programs to be protected by German copyright law. The Directive's definition of originality (which has no counterpart in the U.S. statute, but which is consistent with the U.S. case law interpretation of the originality requirement) was thus needed to harmonize EC law. See CZARNOTA & HART, supra note 2, at 44.


9. Directive, supra note 1, art. 8.1; 17 U.S.C. § 302 (1988). The Directive differs from U.S. law in the duration of protection contemplated for programs whose authors are companies (by virtue of "work made for hire" rules). The Directive provides for 50 years of protection from the date of the first commercial distribution of the program. Directive, supra note 1, art. 8.1. U.S. law provides for 75 years of protection from the date of publication or 100 years from the date of creation, whichever expires first. 17 U.S.C. § 302(c). Since adoption of the Directive, the EC has extended copyright terms to life of the author plus seventy years. See Proposal for a Council Directive Harmonizing the Term of Protection of Copyright and Certain Related Rights, COM(92)33 final. The U.S. Copyright Office has asked for public comments on a proposal to extend the U.S. term in a similar way. The EC and U.S. proposals would also lengthen the copyright terms for works whose authors are companies.
What U.S. and EC laws mean by "computer programs" is less similar than might at first be apparent. U.S. copyright law defines the term "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." The EC Directive does not define the term "computer program." A reason sometimes cited for the EC's omission of a definition of this term is the likelihood that any definition would become outmoded by the advance of technology.

But another reason for not defining the term may have been so that the Directive could protect some things as "computer programs" that would likely be outside the scope of the U.S. definition. The EC Directive expresses an intent to treat both preparatory design materials for programs (such as flow charts) and hardware implementations of software designs as "computer programs" that are subject to the rules set forth in the Directive.

In the United States, a hardware implementation of a software design would probably be beyond the reach of copyright law as a "useful article." Flow charts or other graphic representations of program design elements would likely be characterized under U.S. law as "graphic works" which are subject to the "useful article" limitation on the scope of copyright protection. Although U.S. courts would likely find infringement if a programmer copied elements reflective of a flow chart author's personal style into a competing program, it is also quite possible, given the "thin" scope of protection that U.S. law has traditionally afforded to functional writings, that a program implementing

the functional design depicted in the flow chart would be regarded as noninfringing of the flow chart copyright.\footnote{16}

**B. Special Provisions Applicable to Programs But Not to Other Literary Works**

Also common to U.S. and EC copyright law as applied to computer programs are a number of special rules thought to be necessary to take into account special characteristics of computer programs that distinguish them from other literary works. Both laws provide privileges for making copies of programs necessary to allow program instructions to be executed, for making backup copies of the code, and for making some modifications to program code (such as fixing "bugs" or errors in the code).\footnote{17} Although these commonalities are important, it is worth noting that the fine details of the special privileges under the laws of the United States and the EC are more different than this general statement of similarities might suggest.

The use, backup, and modification special privileges are, for example, available in the EC to all lawful possessors of copies of programs.\footnote{18} Although the National Commission on New Technological Uses of Copyrighted Works (CONTU) recommended to Congress that similar special rules should be available to lawful possessors of programs under U.S. law,\footnote{19} software developers successfully lobbied Congress for a change that would make these privileges available only to owners of copies of computer programs.\footnote{20} Although defendants in some cases have been successful in claiming the benefits of these privileges notwithstanding their apparent status as licensees,\footnote{21} in other cases, the "owner of a copy" restriction has been enforced and licensees have

\footnote{16. See, e.g., Plains Cotton Coop. Ass'n v. Goodpasture Computer Service, Inc. 807 F.2d 1256 (5th Cir.), cert. denied, 484 U.S. 821 (1987). If, however, the flow chart was unpublished, a trade secrecy misappropriation claim could be asserted in the United States against one who implemented someone else's flow chart's design into his or her own software.}

\footnote{17. 17 U.S.C. § 117 (1988); Directive, \textit{supra} note 1, arts. 5.1, 5.2.}

\footnote{18. Directive, \textit{supra} note 1, arts. 5.1, 5.2.}


\footnote{20. See 17 U.S.C. § 117.}

been denied the privileges that the statute grants to owners of copies of programs.  

In addition, the EC Directive forbids the use of contract provisions that would deny some of these special privileges to lawful possessors of programs.  

In the United States, contractual restrictions on the making of backup copies or modifications to programs are more likely to be enforced.  

Although in these two respects the EC Directive can be viewed as more protective of user/consumer interests than U.S. law, in some other respects, U.S. law may provide broader rights to users than EC law does. The U.S. statute seems to have been intended to provide a broader array of user rights to make modifications to programs than was contemplated for EC law.  

If, for example, one needed to adapt a program to make it run on a different machine, that would seem to be lawful under § 117's special adaptation privilege.  

The EC Directive may be more restrictive.  

There is, however, language in the Directive from which one can infer that the scope of permitted modifications under the Directive will include those which are necessary given the intended uses of the program.  

Secondly, U.S. users can always look to the fair use doctrine if they want to make a reasonable use of a program beyond that contem-


23. Directive, supra note 1, Recitals (forbidding contract restrictions on loading the program and correcting errors), art. 5.2 (similar restrictions forbidden as to backup copying).

24. CONTU contemplated contractual restrictions on section 117 privileges, at least as to adaptations of software. See CONTU REPORT, supra note 19, at 13-14.

25. Compare CONTU REPORT, supra note 19, at 13, with Directive, supra note 1, art. 5.2. Concerning the varying interpretations that might be given to the statutory language of section 117 on user modification rights, see, e.g., Pamela Samuelson, Modifying Copyrighted Software: Adjusting Copyright Doctrine to Accommodate a Technology, 28 JURIMETRICS J. 179 (1988). Although an early U.S. decision had given section 117 a restrictive interpretation, see id. at 190-91, later decisions have given it a broader interpretation, see, e.g., Foresight, 719 F. Supp. 1006.

26. See CONTU REPORT, supra note 19, at 13.


29. Id. (expressing the view that a program can be used for its intended purpose).
plated by Congress or by the terms of a licensing agreement.\textsuperscript{30} Most member states of the EC do not have fair use provisions.\textsuperscript{31} In line with the civil law tradition from which it largely sprang, the Directive is grounded on the assumption that all uses of programs for which rules are needed have been addressed in the Directive. Thus, the scope of permitted uses under the Directive is likely to be confined largely to that which has been expressly authorized.\textsuperscript{32} Moreover, the Directive intends to forbid member states from using fair use doctrines (or the like) to expand certain privileges beyond that which the Directive established.\textsuperscript{33}

The U.S. copyright statute, in contrast, was intended to provide a basic framework of rights. Although the U.S. statute contains highly detailed provisions on certain topics, Congress also knew that issues would sometimes arise in litigation for which the statute would provide no ready answer. It expected that the fair use doctrine would provide a way to balance competing interests when new technologies presented questions which the statute did not address.\textsuperscript{34} Indeed, on one important issue as to which the Directive has a highly detailed provision, namely, the lawfulness of decompilation, U.S. courts relied on the fair use provision to achieve a substantially similar result, as the next subsection will show.

C. Decompilation and Internal Interfaces

In the aftermath of two important U.S. appellate court decisions, \textit{Sega Enterprises v. Accolade, Inc.}\textsuperscript{35} and \textit{Computer Associates Interna-}

\begin{itemize}
  \item \textsuperscript{30} 17 U.S.C. § 107. The Ninth Circuit Court of Appeals recently characterized as "verg[ing] on the frivolous" an argument that the special limitation provision for computer software, 17 U.S.C. § 117, so defines the scope of permissible fair use for computer software as, in effect, to displace section 107 for computer programs. See Sega Enter. v. Accolade, Inc., 977 F.2d 1510, 1520-21 (9th Cir. 1992).
  \item \textsuperscript{31} See \textit{INTERNATIONAL COPYRIGHT LAW AND PRACTICE} (Paul E. Geller ed., 1992).
  \item \textsuperscript{32} Article 4 of the Directive sets forth the set of exclusive rights granted to the owner of the copyright in a program. Restrictions on the scope of exclusive rights are set forth in Articles 5 and 6. For a discussion of a few respects in which the Directive is likely to be construed beyond its precise text, see \textit{infra} notes 40-46 and accompanying text.
  \item \textsuperscript{33} Directive, \textit{supra} note 1, art. 6.3. See \textit{CZARNOTA \& HART, supra} note 2, at 75-77.
  \item \textsuperscript{34} See, e.g., Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417 (1984). The implications of \textit{Sony} when new uses are made of copyrighted works in the digital medium are explored in Pamela Samuelson, \textit{Fair Use For Computer Programs and Other Copyrightable Works in Digital Form: The Implications of Sony, Sega, and Galoob}, 1 J. INTELL. PROP. L. (forthcoming 1993).
  \item \textsuperscript{35} 977 F.2d 1510 (9th Cir. 1992) (making of intermediate copies of programs is fair use when necessary to get information to achieve functional compatibility) amended, No. 92-15655.
\end{itemize}
tional v. Altai, Inc., U.S. and EC law are now in relative harmony on two very important copyright issues affecting computer programs. Both laws now consider it lawful, as a matter of copyright law, for a software developer to decompile computer programs (which necessarily involves making intermediate copies of them in seeming violation of copyright's exclusive reproduction right) in order to get access to otherwise unavailable interface information (that is, information about the way the program has been structured to permit it to interconnect with other programs, information which another developer would need to know in order to develop a program that could interoperate with that program). It is also lawful under U.S. and EC copyright laws to reproduce interface information in another program in order to achieve interoperability.

Although the Directive most clearly speaks of interconnection requirements among programs (i.e., program-to-program interfaces), the Directive can fairly be read as regarding requirements for program-to-hardware interoperability as equally unprotectable by the copyright law, a result consistent with the law in the United States as well. Moreover, it is reasonably clear that both laws permit decompilation.

1993 U.S. App. LEXIS 78 (9th Cir. Jan. 8, 1993). See also Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992) (applying Ninth Circuit copyright law to rule that decompilation was fair use); Last Frontier Conference Report on Copyright Protection of Computer Software, 30 JURIMETRICS J. 15 (1989) [hereinafter Last Frontier Report] (expressing a consensus view of ten copyright scholars that decompilation for purposes such as obtaining information needed to develop a compatible program should be regarded as fair use).

36. 982 F.2d 693 (2d Cir. 1992) (reproducing elements of programs required to achieve compatibility with other programs held non-infringing). See also Sega, 977 F.2d 1510 (indicating that elements of programs needed to achieve compatibility are unprotectable by copyright law).


38. Directive, supra note 1, art. 6.1; Sega, 977 F.2d at 1527-28.

39. Directive, supra note 1, Recitals; Altai, 982 F.2d at 709-10.

40. Directive, supra note 1, Recitals (referring to the possibility that physical interconnections may be required). The Sega case seems to have involved a program interface needed to achieve compatibility with hardware. See Sega, 977 F.2d 1510.
and use of interface information not only to develop a program that will interoperate with the decompiled program, but also to develop a program intended to be sold as a competitive substitute for the decompiled program.\footnote{Altai involved the taking of interface information to develop a competing program. \textit{Altai}, 982 F.2d 693. Although some U.S. commentators on the EC Directive have the impression that the Directive only permits development of a program that will interoperate with the program whose interface information is being reproduced, see, e.g., Leo Raskind, \textit{Protecting Computer Software in the European Economic Community: The Innovative New Directive}, 18 Brook. J. Intl. Law 729, 746 (1992), the better view is that use of interface information to develop a competing program is also lawful under the Directive, see, e.g., Thomas C. Vinje, \textit{The Legislative History of the EC Software Directive}, in A HANDBOOK OF EUROPEAN SOFTWARE LAW, supra note 2, at 39.}

Less clear, however, is how the EC Directive would be construed if the future compatibility issue presented in the U.S. case, \textit{Atari Games Corp. v. Nintendo of America, Inc.},\footnote{Nos. C-88-4805 FMS, C-89-0027 FMS, 1993 WL 214886 (N.D. Cal. Apr. 15, 1993).} should arise in the EC. A U.S. District Court judge decided that although Atari Games would not have infringed Nintendo's copyrights had it copied just the interface elements necessary to achieve compatibility with current versions of Nintendo's systems, Atari Games did not have the right to copy any more than this.\footnote{Id. This ruling is consistent with an earlier ruling by the same District Court granting Nintendo's motion for issuance of a preliminary injunction that viewed copying to achieve future compatibility as unlawful as a matter of copyright law. Although the Federal Circuit overturned the trial court's ruling on the decompilation issue, it affirmed the trial court's ruling on the future compatibility issue. \textit{See} Atari Games Corp. v. Nintendo of Am., Inc., 18 U.S.P.Q.2d (BNA) 1935 (N.D. Cal. 1991), aff'd in part, rev'd in part, 975 F.2d 832 (Fed. Cir. 1992).} Atari Games argued that it should have been allowed to copy elements of Nintendo's programs necessary to achieve compatibility with Nintendo units in the event that Nintendo tinkered with its interface so as to make Atari Games' products incompatible. Atari Games offered evidence to show that previous changes in the interface were an unfair competitive "game" at which Nintendo had previously shown proficiency.\footnote{\textit{Atari Games}, 1993 WL 214886, at *6.} Employing the distinctly novel copyright theory that without copyright protection for future compatibility aspects of the interface, Nintendo would have inadequate incentives to develop innovative hardware systems, the District Court rejected Atari Games's future compatibility defense.\footnote{Id. at *7.} Copyright has not generally aimed to be an incentive to investment in hardware.

With the EC's history of litigation with IBM concerning changes to IBM's interfaces that rendered competitive peripheral equipment incompatible—a history that played an important role in the formation
of the EC software directive—the future compatibility issue may be received with greater sympathy in the EC than it has been thus far in the United States.46

In a number of respects, however, U.S. and EC law seem to be at odds on decompilation-related issues. For one thing, the EC Directive is explicit that contractual restrictions on decompilation necessary to achieve interoperability are unenforceable.47 Although it is unlikely that so-called "shrink-wrap license" restrictions on decompilation would be given any legal effect in the United States,48 there is a greater likelihood that negotiated restrictions on decompilation would be enforced in the United States.49 Some commentators have suggested that the copyright misuse doctrine might serve as an equivalent public policy limitation on the ability of copyright owners having substantial market power to forbid decompilation to get access to interface information.50

Secondly, there will almost certainly be a broader scope of decompilation permitted in the United States under the fair use doctrine than the text of the EC Directive would permit.51 The Directive provides for only one set of circumstances in which decompilation is permissible in the EC, namely, where necessary to get access to information for achieving interoperability.52 Even then, the Directive is restrictive about how much of the program can be decompiled and about other uses that can be made of the interface information discerned by study of the decompiled program.53 The only other kind of "reverse engineering" of computer programs that the EC Directive seems to permit is that which can be done by executing program instructions to observe

---

46. The influence of this history was evident from discussions I had with EC policy makers in Brussels to whom I spoke this summer.
47. Directive, supra note 1, art. 9.1.
48. See, e.g., Rice, supra note 22.
49. MAI Systems Corp. v. Peak Computer, Inc., 991 F.2d 511 (9th Cir. 1993) (software license restrictions enforced).
50. See McManis, supra note 37. See Paul Goldstein, Infringement of Copyright in Computer Programs, 47 U. PITT. L. REV. 1119 (1986).
51. Samuelson, supra note 34.
52. Directive, supra note 1, art. 6.
53. Article 6.1 of the Directive authorizes decompilation only as to the parts of the original program which contain information pertinent to interoperability. Article 6.2 limits uses that can be made of interface information discerned through decompilation. The decompiler can use the information to create an interoperable program, but cannot disclose the information to others except as required for their interoperability needs. For a detailed discussion of Article 6 restrictions on decompilation, see Raskind, supra note 41, at 744-47.
and test its functionality, which is sometimes referred to as "black box" testing.\textsuperscript{54}

While a good argument can be made that decompilation should also be permissible under the Directive to the extent necessary for correcting errors in the software, because error correction is an activity permitted under the Directive and decompilation may be necessary to do error correction, this argument would seem to be difficult to defend given how explicit the Directive is about limiting the decompilation privilege to interoperability situations.\textsuperscript{55} It is considerably more likely that the decompilation necessary to be able to modify programs to take advantage of section 117's adaptation privilege would be regarded as a fair use in the United States.\textsuperscript{56}

The \textit{Sega} decision, for example, would seem to make decompiling a program to get access to an unpatented algorithm a fair use as well. The court in \textit{Sega} spoke of decompilation as lawful when necessary to get access to unprotectable elements of computer programs.\textsuperscript{57} An unpatented algorithm would likely be an unprotectable element of a program under 17 U.S.C. § 102(b).\textsuperscript{58} The EC Directive would, however, regard decompilation for this purpose to be infringing, even if this had the incidental effect of permitting program developers to use copyright to protect ideas.\textsuperscript{59}

\textsuperscript{54} Directive, \textit{supra} note 1, art. 5.3. \textit{See} CZARNOTA & HART, \textit{supra} note 2, at 69-70 (discussing the scope of this privilege). Although the EC Directive does not expressly adopt the "black box" model of software copyright protection proposed by Duncan Davidson, a U.S. venture capitalist, the EC Directive is consistent with Davidson's model in a number of respects. \textit{See} Duncan Davidson, \textit{Common Law, Uncommon Software}, 47 U. PITT. L. REV. 1037 (1986). Like the EC Directive, Davidson believes that competitor decompilation to get access to internal elements of software should be regarded as infringing. Davidson, however, views "black box" testing through extensive use of the software as the only appropriate means for reverse engineering programs.

\textsuperscript{55} Directive, \textit{supra} note 1, art. 6.

\textsuperscript{56} Samuelson, \textit{supra} notes 25 and 34.

\textsuperscript{57} That is, the court did not limit the kind of unprotected material to that which was necessary to achieve interoperability. \textit{See} Sega, 977 F.2d at 1525-26.

\textsuperscript{58} Gates Rubber Co. v. Bando Chem. Indus., 28 U.S.P.Q.2d (BNA) 1503 (10th Cir. 1993) (reversing trial court ruling that relied in part on copying of an algorithm, remanding with instructions to consider whether this was an unprotectable element under 17 U.S.C. § 102(b)). \textit{See also} Samuelson, \textit{supra} note 34 (arguing that decompilation to get access to an unpatented algorithm should be noninfringing of copyright).

\textsuperscript{59} A fair use claim made by one who decompiled another firm's program to determine whether it had infringed one's copyright would likely also succeed under U.S. law. \textit{See}, e.g., E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485 (D. Minn. 1985). It is ironic that the EC provides so much protection against decompilation as to put EC developers at risk of not being able to use what may be the best tool technically available for determining whether infringement has taken place.
At this point, it may be worthwhile to step back to understand the context within which the EC Directive arose. At the time EC policy makers were considering what to do about legal protection of computer programs, a complex political struggle arose, both within the EC directorates and among the various lobbyists who sought to influence the content of the software directive. EC policy makers had never seen the likes of the lobbying that took place over the EC software directive, particularly over the decompilation and interoperability provisions.60

There were three principal groups within the EC with differing perspectives on computer program protection issues. A group within the telecommunications and information technology directorate favored *sui generis* (of its own kind) protection for computer programs. A group from the competition policy directorate, which included persons with experience in litigation against IBM over access to interface information issues, believed that interfaces of programs should not be protected by copyright law, and insofar as decompilation might be necessary to get access to interface information, such decompilation should be permitted in order to promote competition in program development. The group actually charged with the task of drafting the Directive was the group in charge of copyright policy issues for the EC. This group tended to favor copyright protection for programs, such as literary works, and tended to be receptive to arguments made by U.S. trade negotiators and lobbyists for some firms that the only special copyright rules needed for software were those for loading software onto a computer, making backup copies, and correcting errors in program code.

U.S. trade negotiators asserted to EC policy makers that under U.S. law, decompilation would probably not be considered a fair use of a computer program copyright because of the commercial and competitive character of the decompilation. Characterizing interfaces as parts of the "structure, sequence, and organization" of computer programs, trade negotiators argued that they were protectable by copyright law under precedents, such as *Whelan Associates v. Jaslow Dental Laboratory, Inc.*61 These views were also forcefully reinforced by lobbyists from a number of major U.S. computer companies, such as IBM, DEC, and Apple. These firms perceived a potential for lost revenues and mar-

---


ket share if EC firms could decompile their system software in order to make interoperable programs without the need for licensing to get access to and permission to use the U.S. firms’ interfaces.

Although at least one prominent U.S. industry group formed to support this “strong” copyright position, U.S. firms were far from united on the scope of protection issues. A number of U.S. firms joined a number of European computer firms to form the European Committee for Interoperable Systems which lobbied in favor of allowing decompilation to get access to interface information and other uncopyrightable elements of programs.62

The eventual adoption of the decompilation-to-achieve-interoperability provisions of the EC Directive represented a political compromise between the “no decompilation and strong protection for interfaces as ‘SSO’” camp and the “decompilation to get access to any unprotected elements but especially to get access to interface information” camp. Each camp had both European and U.S. adherents, and each camp put very considerable energy into getting their way.

Several things are notable about the compromise the EC Directive achieved on the decompilation/interoperability issue. For one thing, achieving it required a shift in perspective for those inured to the Continental European tradition of intellectual property law. In this tradition, competition policy concerns have generally been confined to regulating an owner’s “exercise” of intellectual property rights, but has not been injected into the process of defining the “existence” of an intellectual property right.63 Some traditionalists view this intrusion of competition policy concerns into the formation of a policy on the proper extent of intellectual property rights as an unwelcome development which may have, in the long term, corrosive effects on authors’ rights.64

For the EC to go as far as it did to permit decompilation necessary to achieve interoperability required resisting some very considerable U.S. pressure. One former U.S. trade negotiator still speaks of the Directive as equivalent to an unpaid compulsory license for U.S. firms’

62. See Palmer & Vinje, supra note 60, at 70 (discussing U.S. software industry lobbying on both sides of this controversial issue). The U.S. Trade Representative at the time the Directive was under consideration was Carla A. Hills, long a member of IBM’s Board of Directors. This may help to explain why USTR negotiators tended to have the same views as IBM’s lobbyists on computer software copyright issues.


64. I heard this view expressed in Brussels by some of the European intellectual property professionals with whom I spoke.
valuable interface information. The pressure U.S. negotiators exercised on EC policy makers was, of course, posited on the view that under U.S. copyright law, decompilation—whether to gain access to interface information or for any other purpose—was illegal. One can only wonder what sort of decompilation privilege the EC might have adopted had the Ninth Circuit Court of Appeals rendered its *Sega* decision before the text of the Directive had been finalized.

What is clear is that the EC has frozen in time the political compromise achieved on the decompilation-to-achieve-interoperability issue at a time when the controversy has heated to a high degree. The compromise did what those who made it hoped for: it resolved one very contentious dispute in very precise terms. In doing so, however, it closed the door on further evolution of EC law on decompilation. U.S. courts, by contrast, will engage in further evolution of copyright law as regards decompilation or any other issue which might be litigated by software developers.

Of course, it is precisely because of the evolution in post-Directive U.S. case law that U.S. law now conforms as much to the EC Directive on decompilation and interoperability issues. But since U.S. law will continue to evolve, it appears likely that some of this evolution may in the future cause U.S. case law to diverge from what the EC Directive provides.

D. THE SCOPE OF PROTECTION FOR SOFTWARE INTERNALS

A more important source of potential disharmony between U.S. and EC copyright law affecting computer programs concerns the test courts in the United States and the EC will use to determine software copyright infringement. This, in turn, seems likely to affect the scope of copyright protection that will be accorded to internal elements of programs in these two jurisdictions.

The "successive filtering method" test for software copyright infringement has been gaining an increasing number of adherents in the appellate courts of the United States. This test requires "filtering
out” a broad range of unprotectable elements before the final judgment is made about whether the remaining expressive elements in a program have been copied by a defendant. As yet, it is unclear what test for copyright infringement will be used in the EC. Some believe that use of the successive filtering test would be inconsistent with both the spirit and the letter of the EC Directive; others disagree, pointing to a recent decision in the United Kingdom that seemed to approve of the “successive filtering” method for judging software copyright infringement. Whether the internal aspects of programs will receive a more extensive scope of copyright protection under the law of the EC than in the U.S. may depend on what test for software copyright infringement is adopted in the EC.

Some will argue that the “successive filtering method” is contrary to the intent of the Directive by pointing to the context within which this aspect of the Directive was drafted. At the time the EC was considering what to do about legal protection of computer programs, the seemingly most prominent U.S. case was Whelan. Whelan sets forth a test for software copyright infringement that would have given an extremely broad scope of protection to internal design features of computer programs. Under the Whelan test, only the general purpose or function of a program was regarded as the work’s unprotectable “idea;” all else was “expression” unless there was only one or a very small number of ways to do something, in which case the idea and its expression would be considered “merged” and no protection would be available to the merged element so as not to give a copyright monopoly on an idea. This case was proffered by U.S. trade negotiators and lobbyists for some major U.S. firms as setting the standard for application of copyright law to computer programs to which the EC’s law should conform.

76. One case decided recently in the UK, John Richardson Computers Ltd. v. Flanders & Chemtec Ltd., by the Chancery Division of the English High Court of Justice, has cited Altai and spoken favorably about its successive filtering test for software copyright infringement. Some would argue that this endorsement should not be taken too seriously given that the cause of action in this case accrued before the effective date of the Directive. However, others view the decision’s favorable citation to Altai as signaling a convergence of U.S. and EC law on software protection. See Jonathan Band et al., Computer Associates Crosses the Atlantic and Lake Ontario: Richardson v. Flanders and Delrina v. Troilet, INT’L COMPUTER LAW., June 1993, at 2.

68. 797 F.2d at 1237-38.

69. The Whelan decision has, however, been the subject of intense criticism. See, e.g., Last Frontier Report, supra note 35; Nimmer et al., supra note 66.
Some EC policy makers' views on the scope of protection that should be accorded to software internals were strongly affected by arguments made during the debate about decompilation. Opponents of decompilation asserted that the distribution of programs in object code form in the marketplace was an expression of the developer's intent to maintain human-readable forms of the program as unpublished works in which trade secrets were embodied. Should competitors be free to gain access to these trade secrets through decompilation, they argued, this would have a devastating effect on incentives to invest in the substantial expense of developing complex programs. \(^7\) Moreover, they argued that strong protection for software internals by trade secrecy should be buttressed by a copyright prohibition on decompilation in order to provide proper incentives for investment in software. \(^7\) That this would require use of copyright to protect ideas embodied in the program seemed a necessary incident to providing adequate incentives to invest. \(^7\)

If this bothered any EC copyright traditionalists, it may have been less because copyright law was being used to protect ideas and more because this was another example of copyright being used as an instrument of competition policy. This distresses those steeped in the authors' rights tradition who view the exclusive rights of copyright as deriving from the natural right of authors in their works. \(^7\) Once EC policy makers decided to confer on program developers the mantle of "authorship," and the power of that metaphor set in, \(^7\) it may have seemed natural to view any selection or arrangement of program elements as expressive activity, which copyright law should protect. \(^7\)

If one compares early drafts of the Directive and its final text, one could get the impression that EC policy makers may have decided to broaden the scope of copyright protection for software internals. An

\(^70.\) For a recent restatement of this sort of argument, see, e.g., Arthur R. Miller, Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?, 106 Harv. L. Rev. 977 (1993).

\(^71.\) Id.

\(^72.\) Directive, supra note 1, Recitals (indicating that providing adequate legal protection in order to provide incentives to invest in software development as one of the purposes of the Directive).

\(^73.\) See, e.g., Fakes, supra note 1, at 609-16 (expressing distress at the "moral rights loophole" in the EC Directive).


\(^75.\) This too is consistent with Davidson's "black box" model of software protection in which any discretionary choice of "nonliteral" elements of computer programs and their arrangement should be protected by copyright. See Davidson, supra note 54, at 1082-92.
early draft explicitly excluded algorithms, logic, and programming languages from the scope of copyright protection in programs. The final Directive only speaks of excluding ideas and principles from the scope of copyright protection. It is conceivable that European courts will interpret the "idea" and "principle" language as expressing an intent to exclude fewer internal design elements of software from the scope of copyright.

On the other hand, it can surely be argued that algorithms and logic were intended to be at least partly subsumed under the terms "ideas" and "principles." One explanation heard regarding why the explicit exclusion from protection of algorithms and logic was excised from the Directive had to do with the difficulty of defining such terms. Furthermore, under the national laws of member states—which apply to issues not reached by the Directive—there may be limiting doctrines of copyright that will be construed to limit the scope of copyright protection for program internals within the framework of the Directive's "ideas" and "principles" exclusion. The scope of copyright protection for internal design elements of programs was hotly debated in the EC, and it may be that the Directive "punted" on this controversy, leaving it to courts of member states to resolve the issue. If so, then it may be possible for courts in the EC to decide that an approach to determining software copyright infringement, like that adopted in Altai, may come into use in the EC as well.

If, however, the EC Directive was shaped in part to be in harmony with the state of U.S. software copyright law seemingly represented by Whelan, it is worth noting that EC law would now be disharmonious with U.S. law because the stage of U.S. case law evolution represented by Whelan has come and gone.

The next stage of the U.S. evolution is represented by the Second Circuit Court of Appeals decision in Altai. Although the principal issue in Altai concerned the unprotectability of information needed to construct a compatible program and the case resulted, as noted above, in a ruling consistent with the interoperability provisions of the Direc-

77. Directive, supra note 1, art. 1.2.
78. The only recent decision to have used Whelan's test to give a broad scope of protection to computer program internals was reversed on appeal. See Gates Rubber Co. v. Bando Chem. Indus., 28 U.S.P.Q.2d (BNA) 1503 (10th Cir. 1993).
tive, the Second Circuit's endorsement of the "successive filtering" test for software copyright infringement has much broader implications for the scope of copyright protection available to computer programs in the United States.

The *Altai* test filters out not only elements of programs necessary to achieve compatibility, but also aspects of programs that are to be expected given the kind of program involved, features that have become commonplace in the industry, other things that are not original to the plaintiff or are otherwise in the public domain, and aspects of programs constrained in design by considerations of efficiency or externalities.

In cases that have followed *Altai*'s lead, application of the *Altai* test has also included the filtering out of aspects of programs that are processes, procedures, systems, or methods of operation within the meaning of § 102(b). The Tenth Circuit recently vacated a trial court ruling of infringement that had been based in part on the defendant's use of the same algorithm in its program. The appellate court remanded the case to the trial court with instructions to consider whether this algorithm was a procedure of the sort that Congress had meant to exclude from the scope of protection by enactment of § 102(b). By carefully filtering out these kinds of unprotectable elements before the infringement determination is made, these recent decisions conform to traditional copyright principles and precedents.

*Altai* attacked both Whelan's copyright analysis and also the accuracy of its technical understanding of computer programs. It rejected Whelan's incentive-based argument for extension of copyright protection to virtually all nonliteral elements of programs as inconsistent with the Supreme Court's decision in *Feist Publications v. Rural Telephone*

80. *Id.* at 707-10. In *Altai*, most of the similarities between the two programs at issue were either to be expected in programs of that kind or pertained to that which was necessary to achieve compatibility with other programs. Because neither kind of similarity was protectable by copyright law, the similarities were filtered out by the *Altai* test and no infringement was found. *Id.* at 714-15.


82. The Tenth Circuit's *Gates Rubber* decision correctly observes that the presence of unprotectable elements may be probative on the issue of whether the defendant "copied" something from the protected work, but must be excluded when the trier of fact gets to the ultimate infringement determination as to whether there is substantial similarity to expressive aspects of the plaintiff's work. See Alan Latman, "Probative Similarity" as Proof of Copying: Toward Dispelling Some Myths of Copyright Infringement, 90 COLUM. L. REV. 1187 (1990).


84. *Altai*, 982 F.2d at 705-06.
Service Co. Altai speaks of programs as utilitarian works, indicating that such works have a narrower scope of protection than other kinds of copyrighted works.

The revival of meaningful use of § 102(b) exclusions from protection of "processes," "procedures," "systems," and "methods of operation" should have been predictable, even to observers in the EC, given that Congress had adopted § 102(b) in part to ensure that owners of copyrights in computer programs would not use their copyrights to protect the valuable processes or methods embodied in their programs, but only to protect the programmer's writing.

One can only speculate what the EC Directive might have said about unprotectable elements of programs had the Altai decision been the leading software copyright infringement case at the time the Directive was being drafted, rather than Whelan. Some commentators have predicted that the Altai successive filtering method will come to be used in the EC as well. However, it is not entirely clear that the use of this test is consistent with the intent of the drafters of the Directive. Courts in the EC will have to engage in some common law decision-making to decide whether to give a broad interpretation to the "idea" and "principle" limitations in the Directive or a broad scope of protection to program internals.

E. THE AVAILABILITY OF PROTECTION FOR SOFTWARE EXTERNALS

Although EC policy makers may have decided to give an extensive scope of protection to software internals, there is nothing in the text of the Directive to suggest an intent to extend copyright protection to software "externals," such as features, user interfaces, keystroke sequences, macro systems, or program behavior. From discussions with EC intellectual property policy makers this summer, it was my distinct impression that drafters of the Directive had truly not addressed this set of issues.

85. Id. at 711-12 (citing Feist, 499 U.S. 340 (1991) (rejecting "sweat of the brow" theory of originality and holding the white pages listings of a telephone directory to be uncopyrightable for lack of originality)).
86. Id. at 704.
88. See Band et al., supra note 67.
89. Some commentators, notably Raskind, supra note 41, at 742, have the impression that the EC Directive concerns itself with user interfaces. However, in discussions with EC intellectual property professionals this past summer, I had quite a different impression. It is easy to be mistaken
The EC Directive seems to put programmers to the tasks of figuring out independently which algorithms and data structures to use in their programs and of writing their own code, but it does not seem to posit that programmers cannot make their programs do the same things that other programs do or have the same features. If the EC has adopted a model of software copyright protection akin to the "black box" model once advocated by a U.S. commentator, the EC would likely regard emulation of program externals as noninfringing.

The U.S. case law is still evolving about the extent of protection that should be accorded to software externals, such as user interfaces and behavior. Most of the U.S. cases have thus far dealt with the arrangement of words or pictorial elements on computer screens; when traditionally expressive elements of them have been copied, infringement has been found.

EC policy makers are aware that it may eventually be necessary to issue a new directive on legal protection for software externals in the future. However, this future directive may very well not be a copyright-based directive.

F. Conclusion

U.S. and EC copyright law are in harmony in a number of important respects concerning the legal protection of computer programs. The most obvious harmony arises from the fact that both the United States and the EC have adopted copyright as a form of legal protection for programs. Both have also chosen to classify programs as literary works. A number of the special privileges in the Directive and the U.S.

90. See discussion of Davidson's "black box" theory of software copyright protection, supra note 54 and accompanying text.

91. Davidson, supra note 54, at 1080. One EC official indicated in conversation with me that since the EC was giving such extensive copyright protection to software internals, it should give very little to externals.


94. EC intellectual property policy makers also do not intend for the proposed EC industrial design directive to protect software externals.
copyright statute, such as those for use and backup copying, are also quite similar.

In addition, U.S. case law has evolved into substantial conformity with the EC Directive on the unprotectability of elements of programs necessary to achieve interoperability and the legality of decompilation when necessary to get access to interoperability information. These harmonies should facilitate further investment in, and development of, computer software and trade both in the EC and the United States.

There may not, however, be quite as much harmony in the extent of copyright protection provided for programs by these two jurisdictions as one might infer from this statement of similarities. One source of potential disharmony arises from differences in some seemingly fine details in the EC Directive and the U.S. statute regulating software that may be quite commercially important, such as those defining the scope of user modification rights and provisions of the Directive forbidding certain licensing restrictions that have no counterpart in the U.S. law. A more important potential source of disharmony may arise as a result of differences in the EC Directive and the U.S. statutory provisions excluding certain subject matters from the scope of copyright protection available to programs. The U.S. statute contains a more explicit broad exclusion of functional elements embodied in software, such as processes, procedures, systems, and methods of operation, than the scope of copyright in programs than the EC Directive would seem to do. As indicated above, however, the national laws of member states of the EC may provide a basis for giving a broad interpretation to the “ideas and principles” exclusions in the Directive.

Exacerbating these differences may be the very different civil and common law traditions of these jurisdictions. In conformity with the civil law tradition of a predominance of its member states, the EC’s software directive establishes very precise rules that constitute the law governing that domain. U.S. law, in contrast, evolves on a case-by-case common law basis within general statutory guidelines. Although some of the U.S. case law developments since adoption of the EC Directive have brought U.S. law more into conformity with EC software copyright law, others, in particular, the adoption of the successive filtering method test for software copyright infringement by numerous federal appellate courts, may prefigure a narrower scope of copyright protec-

tion for internal design elements of software in the United States than in the EC. However, this remains to be seen. That an English case has spoken approvingly of the successive filtering method in a software copyright case may mean that EC and U.S. law will converge with respect to the scope of copyright protection for software internals.

There are a number of important software protection issues to which the EC Directive is silent, especially as to the protection of external aspects of software, such as features, user interfaces, keystroke sequences, and program behavior. Courts in the EC will either have to engage in common law decision making themselves or reject claims for copyright protection in these kinds of program externals because of the silence of the Directive on these issues. The United States seems much more likely to employ its common law decision making process to deal with questions about the scope of copyright protection for these aspects of programs.

In short, the laws of the United States and the EC as to the legal protection of computer programs by copyright law are in harmony in a number of important respects, but they may not be in as much harmony as they might seem.