Santa Clara University
School of Law

Santa Clara Law Review

Volume 23 | Number 4 Article 3

1-1-1983

Problems of Interpretation under the 1980 Computer Amendment

J. Clark Kelso

Alexandra Rebay

PROBLEMS OF INTERPRETATION UNDER THE 1980 COMPUTER AMENDMENT

J. Clark Kelso*
Alexandra Rebay**

I. INTRODUCTION

In 1980 Congress enacted legislation amending the Copyright Act of 1976 so as to extend federal statutory copyright protection to computer programs.[1] The status of computer programs under the copyright laws had long been a subject of debate among computer industry representatives, the Copyright Office of the United States, Congress, and many commentators.[2] Now, with the enactment of the 1980 Amendment, the principal issues involve questions of statutory interpretation. The courts have accordingly become the main forum for this debate.[3]

In any copyright action, the plaintiff must prove ownership of a valid copyright in the work whose infringement is alleged and copying of the work by the defendant.[4] If the plaintiff's proof is sufficient, the defendant will be found liable

---

* J.D., 1983, Columbia Law School. Research Assistant to the Dean, McGeorge School of Law.

** J.D., 1984, Columbia Law School.

1. Copyright Act of 1976, Pub. L. No. 96-517, 94 Stat. 3028 (codified as amended at 17 U.S.C. §§ 101, 117 (1982)) [hereinafter cited as 1980 Amendment].

2. The writings in this area are voluminous. Much of the material, however, has become obsolete in light of the amendments to the copyright law discussed in this article and the Supreme Court's recent patent decisions. *See infra* text accompanying notes 26-34. A bibliography of patent-related works appears in Scott, *Bibliography*, 1 COMPUTER L.J. 233 (1978). For references to computers and copyright, see TECHNOLOGY & COPYRIGHT (G. Bush & R. Dreyfuss eds. 1979) and TECHNOLOGY & COPYRIGHT (G. Bush ed. 1972).

3. *See, e.g.*, Apple Computer, Inc. v. Formula Int'l.,Inc., 725 F.2d 521 (9th Cir. 1984); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct 690 (1984); Williams Electronics, Inc. v. Artic Int'l., Inc., 685 F.2d 870 (3d Cir. 1982); Data Cash Sys. v. JS & A Group, Inc., 480 F. Supp 1063 (N.D. Ill. 1979), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980).

4. 3 M. NIMMER, NIMMER ON COPYRIGHT § 13.01 (1982).

unless an affirmative defense, such as fair use, is shown.[5] Each of these three requirements for a successful copyright action presents distinct problems when applied to computer programs as provided by the 1980 Amendment.

The first requirement, ownership of a valid copyright, involves, among other things, a finding that the subject matter of the work is copyrightable.[6] The three main issues of copyrightability of software currently facing the courts are discussed in this article. The first issue is, do computer programs lose copyright protection when they move from one physical manifestation to another (for example, from paper to machine, or from punch cards to tape to disk)? Second, do computer programs lose copyright protection when they are translated from source into assembly language and then into object code? Third, what is the relationship between copyright protection and patent protection, and should the possibility of patentability preclude copyrightability?

The second requirement, copying by the defendant, is usually satisfied by the factual finding of access plus "substantial similarity."[7] This article proposes a functional and objective definition of "substantial similarity" that would further the objectives that led to passage of the 1980 Amendment.

The third requirement concerns the absence of an affirmative defense. The most common substantive affirmative defense is the "fair use" of the copyrighted material.[8] The 1980 Amendment supplements the fair use provision of the 1976 Act. This article provides an interpretation of the new statutory language.

We take as our central thesis the proposition that the 1980 Amendment is intended to redress the economic imbal-

---

5. *Id.* at § 13.05.

6. Nimmer lists the following elements as sufficient to show ownership of the copyright by the plaintiff:

> (1) Originality in the author, (2) copyrightability of the subject matter, (3) citizenship status of the author such as to permit a claim of copyright, (4) compliance with applicable statutory formalities, and (5) (if plaintiff is not the author) a transfer of rights or other relationship between the author and the plaintiff so as to constitute the plaintiff the valid copyright claimant.

Id. at § 13.01[A]. Copyrightability of the subject matter is the only area where serious questions have been raised concerning computer programs.

7. *Id.* at § 13.01[B].

8. *Id.* at § 13.05.

ance and unfairness that would result if computer programs were not protected by copyright. This proposition, combined with the duty of courts to effectuate congressional intent, implies that judicial interpretations that upset the balance by denying copyright protection in cases where it is economically necessary or by expanding copyright protection beyond the point at which it is economically necessary, should be avoided. In this way, court decisions may protect the legitimate economic interests of the copyright owner without unduly harming society's interest in the widespread distribution of ideas.

II. The Legislative History of the 1980 Amendment

Copyright law was recodified by the Copyright Act of 1976.[9] At that time, however, Congress was unable to decide how to treat copyright in relation to computer software and databases. Congress had created, in 1974, the National Commission on New Technological Use of Copyrighted Works [hereinafter the Commission] to study and make recommendations concerning, among other things, the "use of copyrighted works of authorship . . . in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information."[10] Pending the Commission's recommendations, Congress froze the copyright law as it pertained to computer uses of copyrighted works as of December 31, 1977.[11]

The Commission issued its Final Report on July 31, 1978,[12] recommending that the copyright law be amended to

---

9. Copyright Act of 1976, Pub. L. No. 94-553, 90 Stat. 2541 (codified at 17 U.S.C. §§ 101-801 (1976)) [hereinafter cited as 1976 Copyright Act].

10. Pub. L. No. 93-573, 88 Stat. 1873 (1974) (creating the Commission).

11. Section 117 of the 1976 Copyright Act provides:

> Notwithstanding the provisions of sections 106 through 116 and 118, this title does not afford to the owner of copyright in a work any greater or lesser rights with respect to the use of the work in conjunction with automatic systems capable of storing, processing, retrieving, or transferring information, or in conjunction with any similar device, machine, or process, than those afforded to works under the law, whether title 17 or the common law or statutes of a State, in effect on December 31, 1977, as held applicable and construed by a court in an action brought under this title.

The 1976 Copyright Act became effective on January 1, 1978. *See* 17 U.S.C. §§ 301-303. Two courts have had the opportunity to interpret section 117. *See infra* text accompanying notes 56-63.

12. *See Final Report of the National Commission on New Technological Uses*

reflect copyright protection of computer software. In support of its conclusion that copyright protection should be extended to computer programs, the Commission made three findings: (1) to encourage the creation and dissemination of computer programs, it is necessary to give program authors some form of legal protection;[13] (2) means of protection other than copyright are ineffectual or impose too high a cost on society;[14] and (3) copyright will provide sufficient protection for the author without unduly restricting public access to computer programs.[15]

Congress, in the 1980 Amendment, adopted the Commission's recommendations for new legislation with only one change in statutory language and without significant comment.[16] Because the Commission was a creation of Congress

---

of Copyrighted Words, (Library of Congress 1979) [hereinafter cited as *Final Report*] *reprinted in significant part in* 3 COMPUTER L.J. 53 (1981).

13. *Final Report*, *supra* note 12, at 11.

14. *Id.* at 16-23.

15. *Id.* at 12.

16. The following was added at the end of section 101 of the 1976 Copyright Act: "A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (Supp. IV 1980). Section 117 was amended to read:

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

17 U.S.C. § 117 (Supp. IV 1980). The language of the Congressional amendment appears in the *Final Report*, *supra* note 12, at 12.

The Commission's version of section 117 applied to "the rightful possessor" of a copy of a computer program, *Id.*, while section 117 as passed by Congress applies only to "the owner" of a copy of a computer program. Given this change in language, it seems likely that courts will interpret the word "owner" narrowly. In particular, lessees will probably not be covered by section 117, and their right to copy or adapt a program will be governed by the terms of the lease rather than by the copyright law. Thus, a copyright owner apparently can maintain much greater control over use of a program by leasing it instead of selling it.

charged with suggesting statutory amendments, and because Congress adopted the Commission's suggestions, the legislative history of the 1980 Amendment is most appropriately found in the work of the Commission.[17]

This article strives to conform to the following guidelines for statutory interpretation:

> The goal of interpretation is to determine the meaning of a statute's words by asking what they would mean to a reasonable person familiar with the objective context of the statute, including the mischief it was designed to remedy, the temper of the times, and most importantly, the statute's purpose, derived in part from the statute's legislative history.[18]

In accordance with these guidelines, we first examine the findings of the Commission in order to determine the "mischief" the Commission believed existed and how the Commission thought that mischief could best be remedied.

---

17. *See, e.g.*, Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521, 524-25 (9th Cir. 1984); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1247-54 (3d Cir. 1983), *cert dismissed*, 104 S. Ct. 690 (1984).

18. R. KELSO & C. KELSO, STUDYING LAW: AN INTRODUCTION 256 (West 1984). The quotation describes the Holmesian or classical method of interpretation. A challenge to this method, according to Kelso and Kelso, was presented in the 1940's and 50's with the rise of Instrumentalism:

> The instrumentalist goal of interpretation is to find the legislature's intent in order to carry out its purpose. The words of a statute are the starting point for evidence of intent and purpose. But words don't have fixed meanings and therefore all relevant evidence, including legislative history, should be considered to discover the intent and purpose of the legislature. *A court should also strive to interpret the statute to reach a fair or sensible result.*

Id. (emphasis added); *see also* Kernochan, *Statutory Interpretation: An Outline of Method*, 3 DALHOUSIE L.J. 333 (1975). The major difficulty with the instrumentalist approach is that, although it

> puts the most evidence into the hands of the court . . . it allows great weight to be given to the court's inference of legislative purpose, and that inference may be rather distant from the statute's actual words. There is a real possibility that a judge's own notions of policy may become enmeshed in conclusions on what the legislature intended.

R. KELSO & C. KELSO, *supra* at 256. This tendency may be especially pernicious with regard to economic and business regulation where stability and certainty are important factors, as opposed to social regulation where goals of fairness and justice play a larger role. Copyright protection falls into the class of economic and business regulation and, therefore, the Holmesian approach is the more suitable guide to interpretation.

A. *The Necessity of Legal Protection for Software*

Based on its study of the economics of the computer software market, the Commission first found that in order to encourage the creation and dissemination of computer programs, authors must be given some form of legal protection.[19] The Commission found that the computer software industry has high development costs (i.e., high fixed costs) but low marginal costs.[20] Thus, the cost of producing the first program is high while the cost of duplicating that program for future sales is relatively low.[21] In this type of market, the price of a commodity to consumers cannot be based on marginal costs but must, instead, be based on average cost.[22] Computer programs, therefore, require legal protection since software "pirates" could obtain significant competitive advantage by purchasing a developer's software on the open market, reproducing it for general distribution, and then selling them in direct competition with the developer at drastically lower prices.

To illustrate the point, assume that computer programs have no legal protection and that marginal costs are zero.[23] Company A writes a computer program at a cost of $100,000. It anticipates sales of 1,000 programs over the next several years. In order to break even, A sets a price of $100 for purchase of one copy of the program (that is, the average cost per unit). Suppose that company B buys a copy of the pro-

---

19. NATIONAL COMMISSION ON TECHNOLOGICAL USES OF COPYRIGHTED WORKS, ECONOMICS OF PROPERTY RIGHTS AS APPLIED TO COMPUTER SOFTWARE AND DATA BASES, (1977) [hereinafter ECONOMIC REPORT].

20. *See* ECONOMIC REPORT, *supra* note 19, at II-10 to II-12. The Commission looked at the development costs of only one computer system. They found that fixed costs accounted for approximately 25% of the total cost of the system. The Commission included the cost of hardware acquisition in the total cost. Hardware acquisition amounted to 60-70% of total costs. A software vendor might not have significant hardware costs, and thus, in some cases, the fixed costs will be a much higher proportion of total costs.

21. *See Final Report*, *supra* note 12, at 11: "The cost of developing computer programs is far greater than the cost of their duplication."

22. *See* ECONOMIC REPORT, *supra* note 19, at II-2 to II-4. The ECONOMIC REPORT assumes that the price of a computer program will remain the same during its useful life. In a real market, this assumption may not hold, and the price will drop as a program becomes outdated. In such circumstances, the seller would presumably set the price higher at initial release than average cost would require so as to compensate for the fact that the program may steadily decrease in value.

23. In the example in the text, the marginal costs for both firms should be nearly the same. Thus, it does not affect the validity of the example to assume zero marginal costs.

gram. B's total cost is $100 while A's cost was $100,000. B obviously can cut into A's market by cutting the price. B is reaping the benefits originally sown by A. In order to encourage the development and dissemination of programs, some form of legal protection is needed.[24]

B. *The Inadequacy of Current Law*

The Commission's second finding was that means of protection other than copyright are insufficient. The major legal doctrines which might provide suitable protection are patent law, trade secrecy and unfair competition.[25]

1. *Patent Law*

At the time the Commission made its report, the use of patents to protect computer programs seemed a dim prospect. The Supreme Court had on three occasions ruled against the existence of a patent in a computer program.[26] Furthermore, the Commission was concerned that the difficulty and expense of obtaining a patent would preclude widespread use of patent as a tool to protect computer software.[27] The contours of patent law have been changed, however, by the Court's holdings in *Diamond v. Diehr*[28] and *Diamond v. Bradley*.[29] In *Diehr*, a patent was granted for a computer controlled process of transforming raw uncured rubber into cured rubber.[30] In *Bradley*

---

24. Another solution would be for a third party, *e.g.*, the U.S. Government, to subsidize program development. Such subsidy will hardly be forthcoming given current political and economic realities. *See Final Report, supra* note 12, at 11.

25. *See id.* at 16-23.

26. Gottschalk v. Benson, 409 U.S. 63 (1972); Dann v. Jonston, 425 U.S. 219 (1976); Parker v. Flook, 437 U.S. 584 (1978).

27. *Final Report supra* note 12, at 17:

> The acquisition of a patent . . . is time consuming and expensive, primarily because a patentee's rights are great and the legal hurdles an applicant must overcome are high. A work must be useful, novel, and nonobvious to those familiar with the state of the art in which the patent is sought.

28. 450 U.S. 175 (1981).

29. 450 U.S. 381 (1981).

30. The Supreme Court described the patent as follows:

> The process uses a mold for precisely shaping the uncured material under heat and pressure and then curing the synthetic rubber in the mold so that the product will retain its shape and be functionally operative after the molding is completed. . . . It is possible using well-known time, temperature, and cure relationships to calculate by means of the Arrhenius equation when to open the press and remove the cured prod-

an equally divided Supreme Court affirmed the grant of a patent in a firmware program and related hardware elements that gave other computer programs access to otherwise inaccessible scratchpad registers.[31] Nelson Moskowitz has described the new rule of the Patent and Trademark Office in cases follow-

---

> uct. . . . Respondents characterize their contribution to the art to reside in the process of constantly measuring the actual temperature inside the mold. These temperature measurements are then automatically fed into a computer which repeatedly recalculates the cure time by use of the Arrhenius equation. When the recalculated time equals the actual time that has elapsed since the press was closed, the computer signals a device to open the press. According to the respondents, the continuous measuring of the temperature inside the mold cavity, the feeding of this information to a digital computer which constantly recalculates the cure time, and the signaling by the computer to open the press, are all new in the art.

Id. at 177-79.

The case seems to hold only that the use of a computer program in an otherwise patentable process does not transform the process into unpatentable subject matter: "Obviously, one does not need a 'computer' to cure natural or synthetic rubber, but if the computer use incorporated in the process patent significantly lessens the possibility of 'overcuring' or 'undercuring,' the process as a whole does not thereby become unpatentable subject matter." *Id.* at 187.

31. Chief Justice Burger took no part in the decision. The circuit court reversed the Patent and Trademark Office Board of Appeals' denial of a patent application in the following invention:

> Appellants' invention is in the field of computer technology. It does not relate to computer applications, i.e., any specific task that a computer is asked to perform, but rather to the internal operation of the computer and its ability to manage efficiently its operation in a multiprogrammed format. A multiprogrammed format is one in which the computer is capable of executing more than one program, and thus perform [sic] more than one application at the same time, without the need to reprogram the computer for each task it must perform.
>
> Specifically, the invention relates to altering or repositioning information in the computer's system base . . . . Prior art systems altered the system base information resident in the scratchpad registers by either reinitializing the system base (completely new information), a process which consumes a considerable amount of time, or by using software which takes advantage of the model-dependent properties of the particular computer. The latter method has the undesirable effect of resorting to reliance on model-dependent software, which is unacceptable to some computer users.
>
> Appellants' invention enables system base information to be altered without having to resort to these techniques and their accompanying drawbacks. They accomplish their result by employing a "firmware" module, consisting of hardware elements permanently programmed with a microcode, which directs the data transfers, between the scratchpad registers and the system base located in main memory, which are necessary to effect the alteration.

In re Application of Bradley, 600 F.2d 807, 808-09 (C.C.P.A. 1979).

ing *Diehr* and *Bradley*:

> The claims [are] first analyzed to determine if they [recite] nonstatutory mathematical algorithms as defined in [*Gottschalk v. Benson*, 409 U.S. 63 (1972)]. Claimed inventions not containing such algorithms [are] found to be within the statutory categories of invention. If the claims [are] found to recite mathematical algorithms, the claims [are] then considered as a whole to determine if the algorithm [transforms] at least one element of the invention to a different state or thing. If such a transformation [takes] place, the claims [are] considered to be within the statutory categories of invention.[32]

It is clear from the above statement that patent law will provide protection for only a portion of all computer programs. The protection, however, would seem to include an economically important group of programs — those which control machine elements in industrial processes.[33] Nevertheless, this change in patent law does not significantly undermine the Commission's finding, since patents for many computer programs will be impossible to obtain under the new test. Moreover, the patent requirements of novelty, usefulness, and nonobviousness remain intact,[34] and acquisition of a patent is still time-consuming and expensive.

---

32. Moskowitz, *The Patentability of Software Related Inventions After Diehr and Bradley*, 63 J. PAT. OFF. SOC'Y 222, 230 (1981).

33. Patent protection for these types of computer programs would complement Nimmer's view of what copyright law should protect. As discussed more fully below, *see infra* text accompanying notes 87-94, Nimmer suggests that copyright should protect only those programs which produce copyrightable output. Under the current patent law, however, such programs might be adequately protected. The patent in such cases would protect not the computer program by itself, but, rather, the whole invention, i.e., the program plus the other elements of the invention that the program controls. Thus, the patent would not prevent someone from taking the program and using it to control a different set of elements. Such use is, however, most unlikely, since the computer programs which perform these types of functions are usually tailor-made for the particular invention. Therefore, most programs would not be easily adaptable to another environment, and the patent in the whole invention would serve to protect the program itself. It is interesting to speculate what the Commission would have done if it had had the Supreme Court's decisions before it. It may well be that the Commission would have agreed with Commissioner Nimmer in his concurrence that copyright is not the appropriate means to protect computer programs which do not produce copyrightable output.

34. *See* 35 U.S.C. §§ 101-103 (1976).

2. *Trade Secrecy*

The Commission found that although trade secrecy is available for more programs than are patents,[35] trade secrecy imposes too high a cost on society. Maintaining secrecy adds costs to the process of writing computer programs.[36] There is also a cost to society in the duplication of effort among competitors.[37] Further, since a trade secret may be lost by disclosure or reverse engineering, the price to consumers will be higher so as to insure against the possibility of the loss of the secret. Finally, trade secrecy is probably not available for programs designed for mass distribution (for example, computer games).[38] The issue of whether copyright law preempts trade secret law remains unsettled.[39]

3. *Unfair Competition*

A third possible source of protection is unfair competi-

---

35. The subject matter of trade secrets is broader than patent subject matter, and the requirements for securing trade secret protection are easier to satisfy than requirements for securing a patent. The subject matter of a trade secret is described as follows:

> A trade secret may consist of any formula, pattern, device or compilation of information which is used in one's business, and which gives him [sic] an opportunity to obtain an advantage over competitors who do not know or use it. It may be a formula for a chemical compound, a process of manufacturing, treating or preserving materials, a pattern for a machine or other device, or a list of customers.

RESTATEMENT OF TORTS § 757 comment b (1939). The only other requirements are secrecy and novelty. The novelty requirement is less strict than that imposed in patent law, but more strict than the originality requirement imposed in copyright law. *See generally* Bender, *Trade Secret Protection of Software*, 38 GEO. WASH. L. REV. 909 (1970).

36. For example, comments in the source code are omitted to make it more difficult to understand the program; incorrect comments may be added to mislead; the program may be made needlessly complex; nonsensical variable names may be used; or unnecessary lines of source may be added. All of these devices increase costs by increasing complexity and decreasing readability. *See* ECONOMIC REPORT, *supra* note 19, at III-5.

37. *See id.* at III-14.

38. *See Final Report, supra* note 12, at 17.

39. *Compare* Management Science America, Inc. v. Cyborg Systems, Inc., 6 COMPUTER L. SERV. Rep. 921, 926 n.4 (N.D. Ill. 1978) (dicta that state trade secret law was not preempted by federal copyright law in case involving marketing of computer program with a copyright notice under a trade secret license) *with* Avco Corp. v. Precision Air Parts, Inc., PAT. TRADEMARK & COPYRIGHT J. (BNA) No. 496 at A-1 (1980) (action for misappropriation of trade secrets in drawings and specifications preempted by federal copyright law). A bill has been introduced settling the issue in favor of non-preemption. H. R. 6983, 97th Cong., 2d Sess. (1982).

tion, including the tort of misappropriation.[40] Although there is some federal protection under section 43(a) of the Lanham Trademark Act of 1946,[41] unfair competition is primarily a creation of state statutory and common law. Misappropriation is broadly defined. It generally requires proof of three elements:

> (i) the creation of plaintiff's product through extensive time, labor, skill and money, (ii) the defendant's use of that product in competition with the plaintiff, thereby gaining a special advantage in that competition (i.e., a "free ride") because defendant is burdened with little or none of the expense incurred by the plaintiff, and (iii) commercial damage to the plaintiff.[42]

This tort would seem to cover most of the evils contemplated by the Commission. The tort was, however, summarily rejected by the Commission as insufficient to the task.[43] The question now is whether the Copyright Act has preempted misappropriation law with respect to computer programs. The Commission wrote that "unfair competition may provide relief *ancillary* to copyright in certain situations,"[44] but it is far from clear that the Commission's belief will become law. Indeed, one court has already declared that the Copyright Act

---

40. *See Final Report, supra* note 12, at 18.

41. 15 U.S.C. § 1125(a) (1976). A broad reading of the Lanham Act making it nearly co-extensive with state unfair competition law was made in L'Aiglon Apparel, Inc. v. Lana Lobell, Inc., 214 F.2d 649 (3d Cir. 1954). *See generally,* Allison, *Private Cause of Action for Unfair Competition Under the Lanham Act,* 14 AM. BUS. L.J. 1 (1976) (discussion of *L'Aiglon Apparel*).

42. Dannay, *The Sears-Compco Doctrine Today: Trademarks and Unfair Competition,* 67 TRADE-MARK REP. 132, 134 (1977).

43. The Commission wrote of unfair competition that "its scope is not as broad [as copyright's], and it seems unlikely that it alone could provide sufficient protection against the misappropriation of programs. For example, the unauthorized copying of any work for any purpose could be a copyright infringement without amounting to unfair competition." *Final Report, supra* note 12, at 18. The difficulty with the Commission's argument is that it assumes that copyright should be extended to *all* computer programs, regardless of a program's economic importance to the author. This assumption is not warranted and is inconsistent with the economic rationale supporting the extension of copyright to computer programs. *See supra* text accompanying notes 19-24.

The Commission was also concerned that since unfair competition law is largely a state doctrine, there is no national uniformity. *See Final Report, supra* note 12, at 18.

44. *Final Report, supra* note 12, at 18. The Commission does not specify in what fashion such relief would be "ancillary" to that provided by the copyright laws. Perhaps the word means simply that a claim for unfair competition could be joined to a claim for copyright infringement.

does preempt misappropriation.[45]

C. *The Adequacy of Copyright*

The Commission found that, on balance, copyright is the proper mechanism to protect computer programs.[46] Copyright is applicable to virtually all programs and is well suited to mass distribution of programs. Further, copyrights are more easily enforced than patents. Finally, copyrights give protection to the owner without denying the opportunity for others to use and benefit from the owner's work.

It is not clear, however, that copyright in its present form will be sufficient. Commissioner Hersey observes, in his dissent to the Final Report, that the uncertainties which remain in the copyright law might drive computer programmers into greater use of trade secrecy.[47] Further, since copyright does not protect the ideas of the author, it may not be appropriate for many computer programs.[48] Often, much of the value of a program derives from a new idea and not from its particular expression. Despite this limitation, however, copyright is a

---

45. *Compare* Synercom Technology v. University Computing, 474 F. Supp. 37 (N.D. Tex. 1979) (program which was not protectable under copyright law could not be protected by state misappropriation law) *with* Roy Export Co. v. CBS, Inc., 503 F. Supp. 1137 (S.D.N.Y. 1980), *aff'd*, 672 F.2d 1095 (2d Cir. 1982), *cert. denied*, 103 S. Ct. 60 (1982) (unfair competition claim not preempted by copyright law where state law protects against commercial immorality and bad faith).

46. The advantages and disadvantages of each of the possible legal theories are displayed in a chart which appears in the *Final Report*, *supra* note 12, at 19.

47. *See id.* at 34. Commissioner Hersey also argues that evidence of pre-1976 practice indicates that the computer industry has opted overwhelmingly for trade secret protection rather than for the available copyright protection. Hersey's concern is that this state of affairs will continue. The addition of copyright protection will only tighten an author's grip on a program. It will not induce authors, despite the increased security from piracy that copyright presumably provides, to disperse their programs more widely within the computer industry. Thus, the goal of increased communication among computer programmers and increased dissemination of programs will not be realized.

Commissioner Hersey's reliance on pre-1976 evidence is, of course, misplaced. The uncertainties remaining in the copyright law as it pertains to computer programs pale in comparison to the state of the law before the 1976 Copyright Act. *See infra* text accompanying notes 52 & 70. Hersey's predictions concerning future behavior may ultimately prove prescient. However, the issue should not be prejudged. The difference in cost between securing trade secret protection and securing copyright protection may be large enough to cause cost-conscious businesspeople and programmers to opt for the less expensive form of legal protection.

48. *See* 17 U.S.C. § 102(b), which provides that copyright protection does not extend to ideas, procedures, processes, systems, methods of operation, concepts, principles, or discoveries.

welcome addition to the protections available to a program author.

III. The Copyrightability of Software

In order to understand the impact of the changes in the copyright law, some understanding of the following mechanics of writing and using a computer program is necessary. In its earliest states, a program may be in the form of a flow chart. Alternatively, it may initially be written on paper in a computer language, such as Fortran, Cobol, or Basic. More commonly, a program is typed onto punch cards which are readable by a Computer or typed directly into a computer and stored on either tape or permanent disks. The program in this initial form is usually called the "source."[49] The next stage typically involves having the computer "compile" the source. The result of compiling is that the source code is translated into a condensed form which the computer may execute more quickly than it would if it had to read the source code while running the program at the same time. The compiled code is usually represented as a series of numbers. This compiled code may be stored in short term memory, on tapes, on disks, or in read only memories (ROMs). This form of a program is called the "object" code. Both source and object code may be printed out on paper. In the final stage, the computer reads the object code and executes one instruction at a time.[50]

The distinctions between source and object code and the forms which both may take are the focal point for two issues:

---

49. For the purposes of this article, source and flow charts may be treated alike. The differences between these two become more critical as the flow chart becomes less detailed. The Commission indicated that a computer program could be a copy of a flow chart if the flow chart was sufficiently detailed to allow a programmer to write out the program line-by-line. This test is in accordance with a proposal made by the Association of Data Processing Service Organizations (ADAPSO) to amend the copyright law so as to clarify the scope of protection in computer software and related materials. The related materials would include a "program description," defined as "a complete procedural representation in verbal, schematic, or other form, in sufficient detail to determine a set of instructions constituting a corresponding computer program." H.R. 6983, 97th Cong., 2d Sess. (1982).

50. There is an intermediate stage between source and object code where the computer program takes the form of "assembly language." For our purposes, assembly language and source may be treated as equivalent. No one has argued in the recent literature that there is any legally relevant distinction between the two. For more information concerning these technical matters, see Note, *Copyright Protection of Computer Object Code*, 96 Harv. L. Rev. 1723, 1724-26 (1983).

First, what physical manifestations may source and object code take without losing their copyright protection? Second, are both source and object code copyrightable subject matter?

A. *Physical Manifestations of Programs Which Have Copyright Protection.*

The first issue has been explicitly addressed in the 1976 Copyright Act and the 1980 Amendment. Computer programs are protectable in nearly all of their forms, whether on paper, stored on computer disks, or stored in memory. In general, protection extends whenever the programs may be "perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."[51] Prior to the 1976 Copyright Act, it was clear that neither source nor object code was protected when stored on disks or in computer memory. The difficulty arose out of the Supreme Court's definition of "copy" in the piano roll case, *White-Smith Publishing v. Apollo.*[52] *Apollo* stood for the proposition that a "copy" was something that could be perceived by the naked eye. If a machine was required to perceive the work, then the work was not a copy.[53] Since source or object code stored on disks or in computer memory could be perceived only with the use of a machine, it was not a "copy."[54]

The 1976 Copyright Act came close to overruling *Apollo* in its entirety. The definition of "copy" in section 101 was clearly designed to overrule the *Apollo* case.[55] On the other hand, section 117 appeared to freeze the law regarding computer uses of copyrighted works since it was before the 1976 Act, while Congress was contemplating further legislation.[56]

---

51. 17 U.S.C. § 101 (1976) (definition of "copies").
52. 209 U.S. 1 (1908).
53. *Id.* at 17.
54. M. NIMMER, *supra* note 4, § 8.08.
55. The House Report explained the purpose of the new definition as follows:
> It makes no difference what the form, manner, or medium of fixation may be—whether it is in words, numbers, notes, sounds, pictures, or any other graphic or symbolic indicia, whether embodied in a physical object in written, printed, photographic, sculptural, punched, magnetic, or any other stable form, and whether it is capable of perception directly or by means of any machine or device "now known or later developed."

H.R. REP. No. 1476, 94th Cong., 2d Sess. 52, *reprinted in* U.S. CODE CONG. AD. NEWS, 5659, 5665. *See also* M. NIMMER, *supra* note 4 § 2.03(B)(1).
56. *See supra* note 11 for the language of section 117.

Two cases have arisen interpreting section 117. The district court in *Data Cash Systems, Inc. v. JS & A Group, Inc.*,[57] concluded that *Apollo* remained applicable and held that the object code in a ROM was not a copy of the source code.[58] On appeal, that decision was affirmed on different grounds. The circuit court found that the plaintiff had lost any possible copyright protection by failing to affix a proper copyright notice.[59] Thus, the circuit court did not consider whether the object code was a copy of the source code.[60] The second district court to intepret section 117 held, in *Tandy Corp. v. Personal Microcomputers, Inc.*,[61] that a ROM was a copy of the source and concluded that Congress intended to extend protection to computer software in this form.[62] The court asserted that section 117 did not freeze copyright law regarding the definition of the word "copy." Rather, section 117 was concerned only with the use of a program in a computer, and not with duplication of a program.[63] The repeal of section 117 by the 1980 Amendment has put to rest the *Apollo* decision with respect

---

57. 480 F. Supp. 1063 (N.D. Ill. 1979), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980).

58. 480 F. Supp. at 1068.

59. 628 F.2d 1038 (7th Cir. 1980).

60. Nimmer argues that the circuit court's holding implied that the ROM was a copy of the computer program:

> That is, the Court of Appeals held for the defendant on the ground that the plaintiff had published its computer program in the form of a silicon chip without affixing a copyright notice . . . thereby injecting the work into the public domain. Since a copyright notice need only be affixed to published copies or phonorecords . . . and since a silicon chip is clearly not a phonorecord . . . the Court of Appeals must have tacitly assumed that it is a "copy," notwithstanding the contrary construction of § 117 by the court below.

M. NIMMER, *supra* note 4, § 8.08, n.18. Nimmer's reading is strained. In light of the absence of copyright notice, the question of whether the ROM was a copy or not was not before the court.

61. 524 F. Supp. 171 (N.D. Cal. 1981).

62. The court quoted the House Report's explanation for the new definition of "copy," *see supra* note 55, and stated that "the imprinting of a computer program on a silicon chip, which then allows the computer to read the program and act upon its instructions, falls easily within this definition." *Id.* at 173.

63. The court noted that section 117 froze the law only "with respect to the use of [a] work in conjunction with" a computer. *Id.* at 174. A duplication is not used "in conjunction with" a computer; it is simply copying. The court also noted that "any other interpretation would render the theoretical ability to copyright computer programs virtually meaningless." *Id.* at 175. The court's reasoning is not entirely convincing in light of the fact that section 117 of the 1976 Copyright Act anticipated further legislative activity.

to source and object code stored on disks or in short term memory.[64]

Computer programs commonly take two other physical forms to which copyright protection may not extend. First, copyright may not protect a program in its final stage as a series of individual instructions executed by the computer circuitry. The Commission indicated that it was unsure whether a program in this form was a protectable work or an unprotectable process.[65] The relevant question is a factual one: Is the alleged infringer *reading* each instruction, which would be no different than reading the program from memory (impermissible under copyright law), or is the infringer simply *observing* the computer as it executes each instruction and deriving the program from that observation, similar to someone who derives game instructions by watching a game being played.[66] Second, copyright may not protect programs that take the form of etchings on a semiconductor chip.[67] Such chips, unlike ROMs, generally do not contain actual instruc-

---

64. "This issue is now rendered moot by reason of the 1980 amendment to Section 117." M. NIMMER, *supra* note 4, § 8.08, at 8-110.

65. The difficulty arises from the Supreme Court holding in Baker v. Selden, 101 U.S. 841 (1879), which states that the use of a system does not infringe the copyright in the description of the system. Thus, according to the Commission, "copyright . . . protects the program so long as it remains fixed in a tangible medium of expression but does not protect the electro-mechanical funtioning of a machine." *Final Report*, *supra* note 12, at 20. The Commission noted later in the *Final Report* that:

> [o]nly when the program is inserted — instruction by instruction — into the processing element of the computer and electrical impulses are sent through the circuitry of the processor to initiate work is the ability to copy lost. This is true at least under the present state of technology. If it should prove possible to tap off these impulses then, perhaps, the process would be all that was appropriated, and no infringemnt of the copyright would occur.
>
> The movement of electrons through the wires and components of a computer is precisely that process over which copyright has no control. Thus, copyright leads to the result that anyone is free to make a computer carry out any unpatented process, but not to misappropriate another's writing to do so.

Id. at 22. *See generally* Reznick, *Synercom Technology, Inc. v. University Computing Co.: Copyright Protection for Computer Formats and the Idea/Expression Dichotomy*, 8 RUT. J. COMPUTERS, TECH. & L. 65 (1980); Note, *Copyright Protection for Computer Programs Under the 1976 Act*, 52 IND. L.J. 503, 511-514 (1977).

66. 1 M. NIMMER, *supra* note 4, § 2.18[3][a].

67. A semiconductor chip is a product "having two or more layers of metallic, insulating, or semiconductor material, deposited on or etched away from a piece of semiconductor material in accordance with a predetermined pattern . . . intended to perform electronic circuitry functions." H.R. 1028, 98th Cong., 1st Sess.(1983).

tions which are read and executed by a processor. Rather, the logic of the circuits on the chip mirrors the logic of a particular computer program. The chip is like the uncopyrightable building built from copyrightable plans.[68] Legislation is currently before Congress to provide a form of copyright protection for semiconductor chips.[69]

B. *The Copyrightability of Source and Object Code*

The differences between source and object code present further difficulties of statutory interpretation. Source code probably was protected even under copyright law as it existed prior to the 1976 Copyright Act. Although no cases found a copyright in a computer program, the Copyright Office began registering source in 1964.[70] A court deciding a case today under pre-1978 law would almost certainly be willing to accept the judgment of the Copyright Office. House Reports make it clear that in the 1976 Act and 1980 Amendment source is classified as a "literary work" under section 102 of the 1976 Copyright Act, and is thus copyrightable subject matter.[71]

The Commission believed that the 1976 Copyright Act also provided protection for object code and, thus, there was no need to add new language to sections 101 or 102 of the Act.[72] This raises a rather difficult problem of statutory inter-

---

68. *See generally* 1 M. NIMMER, *supra* note 4, § 2.08[D].

69. *Id.* Legislation would "amend Title 17 of the United States Code to protect semiconductor chips and masks against unauthorized duplication." 129 *Cong. Rec.* H201 (daily ed. Jan. 27, 1983).

70. *See* COPYRIGHT OFFICE CIRCULAR 31D (Jan. 1965). A certificate of registration constitutes prima facie evidence of the validity of a copyright. *See* 17 U.S.C. § 410 and M. NIMMER, § 7.16(d).

71. The House Committee Report to the 1976 Copyright Act stated that the definition of "literary works . . . includes computer data bases, and computer programs." H.R. REP. No. 1476 94th Cong., 2d Sess. 116, *reprinted in* 1976 U.S. CODE CONG. & AD. NEWS 5659, 5667. The House Report to the 1980 Amendment indicated that the amendment "has the effect of clearly applying the 1976 law to computer programs." H.R. REP. No. 1307, 96th Cong., 2d Sess. 2, *reprinted in* 1980 U.S. CODE CONG. & AD. NEWS 6492, 6509.

72. *Final Report, supra* note 12, at 21:

> The Commission has considered at length the various forms in which programs may be fixed. Flow charts, source codes, and object codes are works of authorship in which copyright subsists, provided they are the product of sufficient intellectual labor to surpass the "insufficient intellectual labor" hurdle, which the instructions "apply hook to wall" fail to do.

pretation. Since there was no language in the 1980 Amendment which definitively addressed the issue, it is uncertain whether Congress adopted the Commission's interpretation of sections 101 and 102 of the 1976 Act. Of course, if the Commission's interpretation of the 1976 Act is correct as a matter of law, then the question of congressional adoption of the Commission's interpretation becomes irrelevant. At least one commentator, Richard Stern, has written that object code was not protected under the 1976 Act and protection was not extended under the 1980 Amendment.[73] Because of the importance of this issue, his arguments deserve some scrutiny.

The major premise of Stern's argument is that "the statutory definition of copy would seem to be conditioned on human-intelligibility."[74] Since object code is, according to Stern, "unintelligible even to trained observers,"[75] it follows that object code cannot be a copy under the copyright law. Although in theory, as Stern argues, a computer program could be written to reverse-compile object code and transform it into source code, this fact does not imply that the object code is a copy of the source code. After all, a chair may be reverse-compiled to produce a blueprint, but that does not mean the chair is a copy of the blueprint for copyright purposes.[76]

These arguments are similar to those presented by Commissioner Hersey in his dissent to the Commission's recommendations. Hersey argued that object code, in contrast to source code, does not communicate anything to humans. Its sole function is to communicate with a machine (or perhaps simply to control a machine). Hersey distinguishes record players and video-tapes on the ground that the sole function of such devices is to communicate the author's expression. Computers, on the other hand, perform other mechanical or

---

73. Stern, *Another Look at Copyright Protection of Software: Did the 1980 Act Do Anything for Object Code?*, 3 COMPUTER L.J. 1 (1981). Stern's arguments are also considered and rejected in Note, *supra* note 50, at 1729-32.

The Third and Ninth Circuits have held that object code is copyrightable. Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521 (9th Cir. 1984); Apple Computer, Inc. v. Franklin Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 104 S. Ct. 690 (1984).

74. Stern, *supra* note 73, at 11.

75. *Id.*

76. *Id.* at 12.

computational functions.[77]

The fundamental flaw in both Hersey's and Stern's treatment is the erroneous assumption that object code is incapable of communicating to the average computer programmer.[78] In fact, reading object code is not fundamentally different from reading musical notes from a score, and is a skill that is readily acquired by any competent computer programmer.[79] Generally, object code takes the form of a long string of numbers. Each number has a specific meaning to the particular type of machine on which the object code is to be executed. In the instruction manuals that accompany many computers, there is a conversion table for object code numbers to machine function. With this table, it is a trivial matter to read the object code and understand the computer program. Since such tables often accompany compilers as well, it is also usually a trivial matter to convert from object code to source statements. But, of course, it is not necessary to make any conversion at all since, with practice, it becomes possible to read the object code "directly." This occurs in the same manner as when a person learns to speak a new language. With practice, conversion from the foreign language to the native language becomes unnecessary, and the person gradually learns to think in the foreign language and to understand it directly.

---

77. *See Final Report, supra* note 12, at 28-29.

78. Hersey and Stern are not alone in making this erroneous assumption. *See, e.g.*, Data Cash Sys. v. JS & A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979); Pope & Pope, *Protection of Proprietary Interests in Computer Software*, 30 ALA. L. REV. 527, 531 (1979); Comment, *Copyright Protection for Computer Programs*, 47 TENN. L. REV. 787, 799 (1980) (citing *Data Cash*).

Professor Maggs has noted that some programs are so large and complex that it would require years of study of the object code to understand the programs. Maggs, *Computer Programs as the Object of Intellectual Property in the United States of America*, 30 (Supp.) AM. J. COMPUTER L. 251, 256 (1982). This fact does not mean, however, that object code should be denied protection. The same difficulties in comprehension can occur with large and complex programs written in a higher-level language, particularly if the programs lack documentation. Surely no one would argue, however, that such difficulties justify restricting copyright only to those programs which have adequate documentation.

79. One of the authors (Kelso) worked for several years as a system programmer on the PLATO system at the University of Illinois. This work entailed reading object code in order to diagnose software bugs which caused the system to "crash." When the system crashed, the only output produced was a printout of the contents of central memory. This constituted approximately 32,000 60-bit words of memory (every three bits was printed as a number in base 8). Despite the overwhelming amount of information produced, it was usually a straightforward task to track down software bugs.

Although source code is usually more convenient to read than object code, there may be times when it is *necessary* to read the object code, for example, when the source program no longer exists or when there are errors in the compiler. In these cases, the object code is a more reliable source of information than is the source program. Thus, if copyright is to be denied to object code, the reasons for such denial cannot include the supposed non-intelligibility of object code.

Even if object code was not protected under the 1976 Copyright Act, it does not follow, as Stern argues, that the lack of protection continues after the 1980 Amendment.[80] The 1980 Amendment clearly implies that object code is copyrightable. The Amendment defines a "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[81] This definition naturally includes object code which is used "directly" in a computer, as opposed to source code which is used "indirectly" in a computer.[82] Further, section 117 was amended by the 1980 Amendment to provide for the "fair use" of computer programs in the preparation of "copies or adaptations."[83] Although the word "adaptation" is not defined in the Act, it seems apparent from the Final Report that the word embraces the compilation of object code from source

---

80. Noting that the 1980 Amendment does not state that all computer programs are copyrightable and does not expressly refer to the copyrightability of object code, Stern argues that the legislative history contained in the Commission's Final Report is therefore of dubious value in light of the 1980 Amendment's silence. Stern, *supra* note 73, at 12.

81. 17 U.S.C. § 101 (1982); *see supra* note 16.

82. Professor Maggs has made the proper interpretation:

> The words "directly or indirectly" are easier to interpret. An example of a program to be used directly in a computer would be one sold on a memory cartridge meant to be plugged into a home computer game system. A program to be used indirectly in a computer would be a program sold on computer tape in the FORTRAN language with the expectation that the user would copy the program to disk storage and then have the computer translate it to machine language.

Maggs, *supra* note 78, at 258. Maggs' article contains other interesting puzzles concerning the definitions of "statements or instructions," "to be used," "a computer," and "a certain result."

Stern rejects this argument on the ground that there is no legislative history to support the interpretation. *See* Stern, *supra* note 73, at 12. There is, however, no legislative history to rebut the interpretation, and Stern offers no other plausible interpretation of the words "directly or indirectly."

83. 17 U.S.C. § 117 (Supp. IV 1980); *see supra* note 11.

code.[84] Therefore, it may fairly be argued by implication that unauthorized copying of object code constitutes infringement of the copyright in the program.[85] Finally, object code must be protected if the economic balance created by the 1980 Amendment is to be maintained. For the computer program pirate, object code is equivalent to source code.[86] It would seriously

---

84. Concerning the "fair use" exception for "adaptations," the commission noted that copyright law "grants to copyright proprietors the exclusive right to prepare translations, transformations, and adaptations of their work." *Final Report, supra* note 12, at 13. The commission defined "object code" as a "version of a program in which the source code language is *converted* or *translated* into the machine language of the computer with which it is to be used." *Id.* at 21 n.109 (emphasis added). The obvious implication is that object code is at least an "adaptation" and perhaps a "derivative work."

Stern argues that object code is not an adaptation of a source program:

> The [Final Report] recommending enactment of section 117 gave two examples of adaptation of programs. One example was conversion of a program from one higher-level language to another, perhaps from BASIC to FORTRAN. This is clearly a reference to source programs, for only they are written in high level languages. The other example was to add features to the program. Again, this necessarily involves source code. Adaptation is a procedure that can be done only at the source code level. It calls for a human being to revise a program intelligible to him, by modifying an intelligible copy written in source code, not object code.

Stern, *supra* note 73, at 13 (footnotes omitted).

Stern's conclusion here again relies almost exclusively on the belief that object code is unintelligible. As previously discussed, in text accompanying notes 78-79, object code is readily intelligible. Furthermore, Stern is wrong when he says that only source code may be modified. During the author's (Kelso's) tenure at the University of Illinois, *see supra* note 79, he occasionally modified the object code of an active program and also wrote a number of small programs in object code, bypassing the source code stage of program development. Finally, for certain applications, conversion of object code is vastly simpler than conversion of source code. For example, the languages for use with Z80 and 8080 microcomputers contain syntactical differences (Z80 is a more advanced version of 8080). However, the 8080 object code is similar to the Z80 object. That is, object code which runs on an 8080 should, with minor modifications, run on a Z80; this is referred to as "upward compatibility". Thus, it would be simpler to convert the 8080 object code to Z80 object code than to convert the 8080 source code.

85. The implication follows from the opening words of section 117: "notwithstanding the provisions of section 106." That is, absent section 117, such an adaptation would have been an infringement under section 106. This conclusion finds support in the *Final Report, supra* note 12, at 13, where the Commission wrote, in support of the necessity for section 117, that copyright "grants to copyright proprietors the exclusive right to prepare translations, transformations, and *adaptations* of their work" (emphasis added). The commission cites 17 U.S.C. § 106(2) (exclusive right to prepare derivative works) for this proposition. It seems evident, therefore, that the commission believed that an "adaptation" was a "derivative work."

86. The cost of decoding object code is negligible compared to the cost of developing a program from scratch. Williams Electronics, Inc. v. Artic Int'l, Inc. 685 F.2d 870, 877 (3d Cir. 1982) (computer programs are copyrightable).

undermine the purpose of the 1980 Amendment if courts were to ignore this reality.

C. *Congruence of Patent and Copyright*

Commissioner Nimmer, in his concurring opinion in the Final Report, proposed a compromise that falls somewhere between the views expressed by the majority of the Commission, on the one hand, and by Stern in his article and Commissioner Hersey in his dissent, on the other. He argued that at some time in the future, "it may prove desirable to limit copyright protection for software to those computer programs which produce works which themselves qualify for copyright protection."[87]

Under this proposed rule, a program designed for use with a legal retrieval system would be copyrightable, as would a program for a computer game. Both produce copyrightable displays. Nimmer notes that programs which control heating or air-conditioning in a building, or traffic signals, or the flow of fuel in an engine would not be copyrightable since the output of the program would not be copyrightable. He explained his proposed rule as follows:

> The [rule] here suggested appears to me to be consistent with the recognized copyrightability of sound recordings. It sometimes has been argued that while printed instructions tell *how* to do work, computer programs actually *do* the work. But this is also true of sound recordings, which in a sense constitute a machine (the phonorecord) communicating with another machine (the record player). A sound recording contained in a phonorecord does not tell a record player *how* to make sounds which constitute a Cole Porter melody. Rather, it activates the record player in such manner as actually to create such a melody. But Commissioner Hersey has made another and most important distinction. "The direct product of a sound recording, when it is put in a record player, is the sound of music — the writing of the author in its audible form." The point is that the operation of the sound recording produces a musical work which itself is copyrightable. That is sufficient to render the sound recording itself copyrightable quite apart from the separate copyright in the

87. *Final Report, supra* note 12, at 27.

musical work.[88]

Under this theory of copyright of recordings, the evil remedied by an action for infringement is not the physical act of copying the magnetic impulses off of a tape, but the appropriation of the capability of communicating what is on the tape to a potential audience.

While there is a surface appeal to Nimmer's view, there are also some practical anomalies. Consider a program which controls water temperature in a washer by opening and closing hot and cold water valves. Such a program, under Nimmer's rule, would not be protected. Now suppose that the program also controls a display panel on the front of the washer. The display panel shows the water temperature and the degree to which the hot or cold water valves are open (for example, "wide open," "half-way," or "shut down"). Such a display might be copyrightable.[89] Does the existence of the display panel imply that *all* of the underlying program is copyrightable, or only that portion of the program which controls the display panel? This difficult question troubled the majority of the Commission.[90] They saw Commissioner Nimmer's proposal as invoking a distinction based solely on the intended use of a program, a distinction not previously made in copyright law.[91]

---

88. *Id.*

89. Copyright might inhere in the format of such a display and perhaps the words used. Professor Maggs also uses a washing-machine example in Maggs, *Some Problems of Legal Protection of Programs for Microcomputer Control Systems*, 1979 U. ILL. L.F. 453, 462-63.

90. *See Final Report, supra* note 12, at 21: "[T]he likelihood is that entrepreneurs would simply require that programs produce a written and, by that token, an unquestionably copyrightable version of their output to obtain copyright in the programs themselves."

Nimmer's response to this objection is "[t]he fact that such a program might also provide for a printout of written instructions (which would be copyrightable) would only render protectable that particular aspect of such a program." *Id.* at 27.

Nimmer's response seems incomplete. He provides no clue as to how a court might go about determining what "particular aspect" deserves protection. Does he mean that only those program lines which contain copyrightable material should be protected? Such an interpretation, protecting only the output and not the computer program itself, seems too narrow. Or, alternatively, does Nimmer believe that the behavior of the entire program is necessary to determine the copyrightable output? This interpretation seems too broad. It is not clear that a reasoned line can be drawn between these two extremes. *See infra* note 93.

91. The Commission wrote that Nimmer's proposal "does not square with copyright practice past and present, which recognizes copyright protection for a work of authorship regardless of the uses to which it may be put." *See Final Report supra*

Although the majority's cursory treatment of Nimmer's proposal may have been warranted in light of Nimmer's own brevity, Nimmer's ideas, when combined with the Supreme Court's recent decisions in *Diehr* and *Bradley*, suggest an interesting congruence between patent and copyright protection of software,[92] namely, that when the primary effect of a program is to control machine elements within a larger mechanical or chemical process, that program should not be copyrightable.[93] Even if the program produces displays for human

---

note 12, at 42. *See* Mazer v. Stein, 347 U.S. 201, 218 (1954); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1252 (3d Cir. 1983), *cert dismissed*, 104 S. Ct. 690 (1984). The majority, despite their rejection of his proposal, seemed to sympathize with Nimmer's concerns: "Although [Nimmer's rule] tries to achieve the separation of idea from form of expression, that objective is better achieved through the courts exercising their judgment in particular cases." *Final Report*, *supra* note 12, at 43.

92. *See supra* text accompanying notes 27-34. Professor Maggs noted the possible congruence of patent and copyright law in an article written after the *Final Report* was issued but before Congress had taken any action on the report: "If Congress passes the legislation recommended by (the Commission), it should exclude from copyright protection computer programs with purely mechanical functions. Concurrently, Congress should provide patent protection for those programs that meet the standards for patents." Maggs, *supra* note 89, at 462.

93. *See supra* note 33. The key words in this statement are, of course, "primary effect." Factors in determining "primary effect" should include such things as the purpose for which the program was originally designed, the use to which it actually is put, and the extent to which human intervention is required for the program to operate within the overall mechanical or chemical process.

An alternative contemplated by both Nimmer and Maggs is that copyright would protect the display or message portions of a program and patent would protect the rest of the program. *See supra* note 90 and Maggs, *supra* note 89, at 462.

As already noted, there is no easy line between display and non-display portions of a program. *See supra* note 90. Assume that the display is a piece of text which physically appears in the program. In such a case, it could be argued that the "display portion" of the program is simply the text itself rather than the program statements which cause the text to be displayed. But, suppose that the program has a dictionary of words with which it may build several sentences. The individual words would hardly seem copyrightable. The copyrightable expression would be the sentences, not the words individually, although the sentences would not appear physically in the program. Another test might, however, include most of the program. The fundamental problem with any of these approaches is that the courts are not technically competent to make the necessary judgments. Further, the distinctions to be drawn involve considerations about which expert witnesses could differ.

This discussion refers only to copyright protection of the program. Even assuming that the program is not copyrightable, it may well be that the display produced by the program is copyrightable. *See, e.g.*, Atari, Inc. v. North American Philips Consumer Electronics Corp., 672 F.2d 607 (7th Cir. 1982) (preliminary injunction granted for infringing copy of PAC-MAN audiovisual game based on characteristics of display); Stern Electronics, Inc. v. Kaufman, 523 F. Supp. 635, 639 (E.D.N.Y. 1981) (preliminary injunction granted for infringement of plaintiff's electronic video games

monitoring of the process, such displays are not central to the performance of the computer program. Nor would production of such displays have been the efficient cause for the creation of the entire computer program. Although not eligible for copyright protection, the program, in conjunction with the machine elements which it controls, should be eligible for patent protection under *Diehr* and *Bradley*.[94] This distinction between copyrightable and patentable computer programs might satisfy the concerns of those who believe that copyright should not be extended to computer programs because of the "machine quality" of computer programs.

IV. SUBSTANTIAL SIMILARITY

The gravamen of any infringement suit is the allegation that the defendant copied the plaintiff's work. Since actual copying is difficult to prove, the courts have developed the doctrine that copying may be inferred if the plaintiff can show that the defendant had access and that the infringing work is substantially similar to the plaintiff's work.[95] Expert testimony may be used as evidence of similarity between two programs. A drawback of the use of expert testimony, however, is its reliance on the subjective beliefs of one person. It is, therefore, advisable to butress subjective expert testimony with ob-

---

"Kamikaze" and "Astro Invaders"). The court wrote in *Stern* that "[a]n audiovisual display is an appropriate subject for a copyright even if the underlying computer program is not copyrighted." *Id.* at 639.

Finally, the proposed interpretation of copyright law is inconsistent with the clear intention of the Commission embodied in the majority's rejection of Nimmer's proposal. Statutory amendment is possible, although unlikely in the immediate future. The only option, therefore, is to attack the constitutionality of the copyright law as it applies to programs whose primary function is to control machine elements. *See* Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812 (E.D. Pa. 1982) (preliminary injunction denied on ground that object code might not be copyrightable on constitutional grounds that it is not a writing), *rev'd*, 714 F.2d 1240 (3d. Cir. 1983), *cert dismissed*, 104 S. Ct. 690 (1984). Such a constitutional challenge is likely to fail, however, since the Supreme Court has historically given Congress wide leeway in defining what constitutes a "writing" for copyright purposes. *See, e.g.*, Goldstein v. California, 412 U.S. 546 (1973); Burrow-Giles Lithographic Co. v. Sarony, 111 U.S. 53 (1884). *See also* Maggs, *supra* note 89, at 461.

94. The overlap between copyright and patent would not be perfect since the hurdles to securing a patent are greater than those to securing a copyright. *See supra* text accompanying note 27. Further, the patent in such a case would protect not the naked computer program, but the whole process of which the program is a part. *See supra* text accompanying note 33.

95. *See* M. NIMMER, *supra* note 4, § 13.03.

jective and provable facts.[96] In order to determine what objective criteria might exist, it is necessary to consider briefly some of the methods of copying computer programs which are likely to be used in the near future.

Exact copying, where the infringing program is identical to the plaintiff's program is the most obvious example of substantial similarity. This form of infringement, is the most economically favorable to the infringer since there will be no need to develop or test the infringing program. Exact copying is thus likely to be a common form of infringement.[97]

The next most obvious example of substantial similarity involves minor changes to such things as the copyright notice, title page, or other identifying parts of the program. This type of infringement is similar to exact coyping. As with exact copying, the cost involved in making such trivial changes is small since there need be no further development or extensive testing of the changes.[98]

While these two examples provide clear cases of substantial similarity, proof of this requirement is likely to be less clear in other cases. In such cases the derivative program may not look like the original program, even though the two programs behave identically. This situation may occur when: (1) all variable names are changed,[99] (2) the source is reordered, i.e., sections or blocks of code are moved around without

---

96. Expert testimony has had a somewhat checkered history. The current thinking seems to be that expert testimony should be allowed if it is the best evidence available: "It has been suggested that '[l]ike the hearsay and original documents rules [the opinion rule] is a "best evidence" rule. The more concrete description is preferred to the more abstract' because it is more reliable, since with each abstraction by the witness the possibility of error increases." J. WEINSTEIN & M. BERGER, WEINSTEIN'S EVIDENCE 701-01 (footnotes omitted). The primary difficulty with expert testimony in infringement actions is that the court is not likely to have the technical competence to choose between the testimony offered. Consequently, "[i]f the plaintiff could find one expert who was willing to testify to striking similarities, the defendant could, no doubt, get two to swear to the contrary." Gemignani, *Legal Protection for Computer Software: The View from '79*, 7 *Rut.* J. COMPUTERS TECH. & L. 269, 288 (1980).

97. Data Cash Systems, Inc. v. JS & A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979).

98. Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171 (N.D. Cal. 1981).

99. In programs written in "higher" languages, such as FORTRAN or PASCAL, the programmer may declare variables with ordinary names. For example, in a computer diary program, important variables might be "day," "month," and "year." If someone were to steal that program, one easy way of altering its appearance is to change the variable names to "dnumber," "mnumber," and "ynumber."

changing the program behavior, (3) minor syntax changes are made which do not change the program behavior, or (4) the program is translated line by line into a different computer language. Each of the above derivations could be carried out almost exclusively by "translator" computer programs which are already in widespread use.[100] These forms of copying are illegal without the consent of the copyright owner.

There are legitimate uses of a copyrighted program in the creation of a new program. Since copyright does not protect ideas, a programmer could use the ideas underlying the copyrighted program. Further, programmers often draw upon solutions to specific programming problems by using nearly identical source statements taken from already tested and working computer programs. This occurs in much the same way that lawyers, when writing new contracts, draw upon the language used in previous contracts.[101] This type of use should be encouraged since it advances the state of the art and does not give the author of the derivative program an unfair economic advantage over the copyright owner. The author of the derivative program cannot avoid costly development and testing of the program.[102] Because the economic imbalance which justi-

---

100. Such programs are currently used for the legitimate purpose of translating a computer program from one language to another. Such translations, if essential for use, are protected by the fair use provisions of section 117. *See infra* text accompanying notes 103-106.

101. In Continental Casualty Co. v. Beardsley, 253 F.2d 702 (2d Cir. 1958), the court held that the defendant's use of the plaintiff's copyrighted legal forms was not an infringement because "[t]he evidence here shows that [the defendant] insofar as it has used the language of [the plaintiff's] forms has done so only as incidental to its use of the underlying idea." *Id.* at 706.

This rule would not protect wholesale copying of an entire program, since there are almost always many ways to write a program. *See Final Report, supra* note 12, n.106. It should be noted, however, that as the size and complexity of a program decreases, the likelihood that several programmers will independently write source code which look alike increases. The degree of similarity will naturally depend on several subjective factors such as the sophistication of the programmers, their previous experience in writing programs directed at the same sorts of problems, and their training in school concerning the style of programs. One important objective factor, is the degree to which a problem admits of a most efficient solution. Most programmers strive to write code that is both elegant and efficient. To the extent that such a goal is attainable, the number of programming solutions to a problem decreases as less efficient or less elegant code is discarded.

102. The use affords some economic advantage to the user since it saves the user from having to develop that particular piece of code independently. However, as long as the bulk of a program is new code or borrowed code put together in a new way, such economic advantage will not be of the character or magnitude required to give the user an unfair advantage. *See supra* text accompanying notes 19-24.

fies copyright protection for the original program does not exist, copyright law should not interfere to prevent the second author's use.[103]

The above considerations and examples suggest that objective indicia of copying include the extent to which one program behaves in the same manner as, or performs the same functions as, the original program,[104] and the ease with which the infringing program may be derived from the original program. We propose that a presumption of copying should be raised against the defendant if the plaintiff can prove the following facts: (1) access; (2) that the allegedly infringing program behaves in the same manner as or performs the same functions as the copyrighted program; and (3) that the allegedly infringing program may be derived from the copyrighted program by a mostly mechanical, inexpensive process.

Application of this test should encourage the development of better programs by allowing programmers to follow the teachings of existing programs without causing severe economic harm to the copyright owner. This result would be consistent with the economic rationale of the 1980 Amendment.

V. Fair Use and Rights of Adaptation

Section 117 was amended in 1980 and now contains provisions dealing with fair use and transfer of programs from one purchaser to another. Legitimate uses include inputting the program into a computer, running the program, and archiving the program. This section clarifies the obvious—the owner of a computer program has the right to use it in his or her computer regardless of the copyright.

There are extremely troubling ambiguities in this section. Section 117 provides, in pertinent part:

> Notwithstanding the provisions of section 106, it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

---

103. *See supra* text accompanying notes 19-24.

104. In all of the examples in the text, the derivative program would behave exactly as the original program. That is, a person using the program would be unable, on the basis of use alone, to determine which program was executing. Identical behavior could also, of course, result from the appropriation of ideas rather than expression. Such appropriation would not be in violation of the copyright law. Thus, identical behavior by itself is not sufficient to support a finding of infringement.

> (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program and that it is used in no other manner . . . .[105]

The first difficulty is determining what constitutes an "adaptation." The Commission described the adaptation right as allowing the program owner the "right to make those changes necessary to enable the use for which it was both sold and purchased."[106] This right includes "[the] conversion of a program from one higher-level language to another to facilitate use . . . [and] *the right to add features to the program* that were not present at the time of rightful acquisition."[107] The right to translate from one high-level language to another makes sense in light of the non-standardization of computer languages. The right to add new features is more surprising. The Commission analogizes this right to that of note-taking in the margin of a book, where even if note-taking is considered to be creating a derivative work, the copyright owner is hardly concerned with such a "violation."[108] The Commission also noted that sales of many software programs are made with full awareness that the program owners will modify the programs. Indeed, some vendors provide technical assistance to users who wish to modify their programs. Finally, to avoid potential injury to the copyright owner, the right to adapt is tempered by the final clause of section 117 which prohibits the program owner from transferring an adaptation without the copyright owner's permission.[109]

Despite the Commission's laudable intentions in fashioning this practical adaptation right, section 117 leaves a number of questions unanswered. Consider the case of educational programs. Most of these programs have the name of the author on a title page, complete, with copyright notice. Does section 117 give a program owner the right to alter the substance of the educational program in preparation for student use? If the materials were in book form, such alteration would violate the copyright owner's right to make derivative works.[110] The copyright owner should, therefore, receive similar protection

---

105. 17 U.S.C. § 117 (1982); *see supra* note 16.
106. *Final Report*, *supra* note 12, at 13.
107. *Id.* (emphasis added).
108. *Id.*
109. 17 U.S.C. § 117 (1982); *see supra* note 16.
110. *See* 17 U.S.C. § 106(2) (1982).

from alterations of a computer program. Section 117 says that such copy or adaptation must be used "in conjunction with a machine, and *that it [may not be] used in [any] other manner.*"[111] A substantive adaptation, in contrast with a technical modification which does not alter what is seen on the screen, would arguably be used not only "in conjunction with" a machine, "as permitted by section 117, but also "in conjunction with" the students taking the materials.[112]

The statute seems overly vague in regard to this problem. While the Commission indicates that the adaptation right "could only be exercised so long as [it] did not harm the interests of the copyright proprietor,"[113] this apparently broad protection of the copyright owner's interests is not reflected in the language of the statute. Given the ambiguity, courts should go beyond the language of section 117 and look to the Final Report by the Commission for guidance as to proper interpretation.

Another problem with section 117 is the lack of guidance regarding what constitutes a fair use of copies or of adaptations of a program.[114] Consider the following facts and hypotheticals: the owner of a copyright in a computer program sells one copy of the program to Buyer. The program is in object code format and is on a diskette. Buyer has ten employees, each of whom has a terminal. The employees use the program in the following ways:[115]

(1) Buyer allows the employees to use the diskette, one employee at a time. An employee puts the diskette into his or her desk-top terminal, and the terminal copies the program from the diskette to a fast memory contained in the terminal. This copy is not an infringing copy since it was created "as an

---

111. 17 U.S.C. § 117 (1982) (emphasis added); *see supra* note 16.

112. Another possible interpretation of the statute which would prohibit such alterations is that the adaptation right is applicable only to computer programs, and not to data which is stored in a computer. Arguably, the words which appear on the screen of a terminal are data and not "statements or instructions" which make up a computer program. Indeed, some computer programs are mere "drivers" that do nothing but pull text out of a data block. If such text is data, then changes to the text would not be protected by the fair use provisions of section 117.

113. *Final Report*, *supra* note 12, at 13.

114. We now focus on the following language of section 117: "(1) that such a new copy or adaptation is created as an *essential step* in the utilization of the computer program in conjunction with *a machine* and that it is used in no other manner . . . . " 17 U.S.C. § 117 (1982) (emphasis added). *See supra* note 16.

115. *See generally* Maggs, *supra* note 78, at 261-262.

essential step" in the use of the diskette.[116] When an employee is finished, the diskette is passed on to the next employee. This procedure is analogous to passing a book around among employees and is not an infringing use.[117]

(2) Buyer makes ten copies of the diskette and distributes the copies to his employees. Each employee may use the diskette in his or her desk-top terminal. This is an infringement of the copyright—it is analogous to making photo-copies of a book for use by employees.[118]

(3) Buyer's ten desk-top terminals are connected by data lines to a central database. The desk-top terminals have their own processor and some fast memory, but nothing is stored permanently in the terminal. In this system, when no one is using the program, the program is located on a single permanent disk (i.e., the central database). When a user desires to access the program, a copy of the object code is sent from the database to the terminal and is there executed. Thus, if all ten employees desire to access the program, there may be ten copies of the program located in the fast memory of the ten terminals.

In this last example the first question is whether the object code stored temporarily in the terminals is a "copy." In order to be a "copy" under section 101, a work must be "fixed" and be capable of being "perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device." To be "fixed," a work must be "sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."[119] Under these definitions, the program stored in memory would seem to be a "copy" since the object code may be perceived with the aid of a machine and

---

116. This is one of the most obvious cases of fair use. A contrary result would be analogous to preventing the owner of a book from reading the book without getting permission from the copyright owner. The Commission wrote the following: "Because the placement of a work into a computer is the preparation of a copy, the law should provide that persons in rightful possession of copies of programs be able to use them freely without fear of exposure to copyright liability." *Final Report, supra* note 12, at 13.

117. See 17 U.S.C. § 109(a) the right of an owner of a particular copy to dispose of that copy; *see also* M. NIMMER, *supra* note 4, § 8.11-.12.

118. Professor Maggs suggests that this result can be reached by arguing "that only the copies necessary for use in the first computer were 'essential' to the use of the program." Maggs, *supra* note 78, at 262; *see infra* text accompanying note 121.

119. 17 U.S.C. § 101 (1982).

will continue to exist in memory for as long as the user desires to use the program—a period of time greater than transitory duration.

The second question is whether the object code is a copy created "as an essential step in the utilization of the computer program in conjunction with a machine" and is, thus protected by section 117. We suggest that such use is not "essential . . . in conjunction with a machine." It is relatively clear from the Final Report that the Commission adopted the language of section 117 in response to the fact that a program in source format usually must be translated into object code in order to run on a computer. The Commission wanted to make clear that this object code is not an infringing copy of the source code since the object code in this example is used solely in conjunction with the use of the program in a machine.[120]

Returning to the hypothetical, it may be argued that the copies in the desk-top terminals are not being used *solely* in conjunction with a machine, but are also being used to allow ten users to access the program simultaneously. Such use is similar to that described in the second example where the program owner made ten copies of the diskette. It is not in conjunction with a machine, but in conjunction with ten users. Thus, it may be argued that the ten copies are infringing since they are not covered by the fair use exception of section 117. Such an interpretation requires adding the following gloss to the statutory language: "as an essential step in the use of a machine *with one user.*"[121] Addition of this gloss is consistent with the intent of the Commission to limit the applicability of section 117 to cases where it does not prejudice the copyright owner.

(4) Now, assume that Buyer has ten terminals which are connected to a time-sharing system. In such a system, there may be only one central computer with a number of terminals connected to it. The computer processes each user for a short time and moves on to the next user. With a fast computer, it

120. *See supra* text accompanying note 84.

121. Alternatively, the words "a machine" may be interpreted to mean "one and only one machine," and "machine" may be defined in such a way that the system described in the text constitutes ten machines (perhaps on the ground that there are ten processors). *See* Maggs, *supra* note 78, at 261. This interpretation, however, leads to difficulty in the fourth and fifth examples to follow in the text.

is possible under this scheme to have hundreds of users accessing programs at virtually the same time (strictly speaking, of course, the users do not access programs "simultaneously" since the computer only processes one program at a time). In Buyer's system, when a user wants to access a program, the computer retrieves a copy from permanent storage and puts it into a fast memory. Each user has his or her own copy of the program in memory. Thus, if all ten employees call up the copyright owner's program, there will be ten copies of the program in memory. We suggest that this scheme is exactly analogous to the desk-top terminal scheme described in the preceding paragraph. The only difference is that instead of having ten copies of the object code residing with each terminal, there are ten copies residing in a centralized memory.

(5) Finally, assume that Buyer has a time-sharing system similar to the one described above. In this system, however, there is only one copy of the program in fast memory regardless of the number of users accessing the program. This is the usual type of time-sharing system because it achieves a savings in the amount of fast memory which is being used by a number of users. The fact that there is only one copy of the object code in memory does not mean that the ten users would see exactly the same displays on their screens. Each user is allocated his or her own set of user-variables which identify where that user is in the program and control the manner in which a particular user sees the program. Are there any infringing copies? As a practical matter, this system is no different from the time-sharing system described in the preceding paragraph. In both systems, ten users will be able to access the same program at virtually the same time. In the former, however, there are ten copies of the object code while in the latter, there seems to be only one copy of the object code.

The first question here is whether, regardless of section 117, this type of use should be considered infringing. Because there are no ordinary fact patterns which are analogous,[122] we

122. There are two characteristics of the example in the text. First, the copyrighted work is being used by more than one person simultaneously. This is similar to a radio or television broadcast. Second, each user may be at a different place in the program. This is unlike a radio or television broadcast where the entire audience receives the same broadcast at the same time. It is this second characteristic that makes conventional analogies difficult to find.

have devised the following: assume ten people are sitting in chairs forming a closed circle. Someone brings in a book which has ten book marks on the first page (the book marks being different colored paper clips). The first person reads for a short time, places the book mark next to the line at which he stopped reading, and passes the book to the person next to him. Each person repeats the procedure. This is surely not an infringing use—there is only one copy of the book. The fact that the readers use this odd interlaced reading pattern is of little consequence as far as copyright laws are concerned.

This interlacing is precisely what takes place in a time-sharing computer system. There is, however, a significant difference between the two examples. In the book example, it will take the readers a longer time for all of them to read the book using the interlace method than if they used a sequential method.[123] In the time-sharing computer system, however, it will take ten users the same amount of time to run all their programs as it would take one user to run a single program. This is due to the fact that even though the computer is serving ten users simultaneously, it is fast enough to serve each user as quickly as desired.[124] This difference in speed means that the use of a time-sharing system is a special benefit to Buyer, a benefit which would not accrue to the buyer of a book under the interlace reading method. Further, this benefit is to the direct detriment of the copyright owner since, but for the time-sharing, the copyright owner might have made ten sales rather than one.

Such use may be deemed infringing by adopting the gloss already suggested,[125] and determining that the copy was not created for use by a machine with one user. While the result may seem counter-intuitive since there appears to be only one copy of the computer program in memory, in practical terms there are ten copies of the program. They may be discerned by realizing that the user-variables are an integral part of the program. Since there are ten user-variables, there are ten different programs running. Although this is a rather subtle ar-

---

123. This is because of the time spent passing the book from one reader to the next. In the sequential method, the book is passed 9 times. In the interlace method, the book will be passed many more times.

124. On the PLATO system at the University of Illinois, for example, up to 400 users may be operating simultaneously without noticeable delays in processing.

125. *See supra* text accompanying note 121.

gument, it reflects the economic realities of computer use. Assume that Buyer in our example leases time on the time-sharing machine. If one company with ten terminals uses one program, the bill to that company will be computed by adding up the time of each user. Assuming each user was on the system for the same length of time, Buyer will receive (10) x (# time units on system) x (price/time unit). The copyright law should recognize this fact and give protection to the copyright owner on the basis of the numbers of potential simultaneous users.

VI. Conclusion

We have suggested in this article that courts should interpret the copyright law as it relates to computer software in accordance with the economic imbalances justifying protection of software by copyright. Thus, to the extent allowed by the statutory and constitutional provisions, courts should not deny copyright protection to programs in object code format if the denial would undermine the economic protection which copyright provides. At the same time, courts should consider the possibility, on constitutional grounds, that programs which qualify for patent protection may be ineligible for copyright protection since copyright protection would be duplicative and serve no economic need. Further, when determining whether one program is substantially similar to another, courts should employ an objective standard that focuses on the economics of copying. Finally, the "fair use" provisions must be interpreted so as to protect the copyright owner's legitimate expectation of remuneration for his or her efforts while, at the same time, protect society's interest in the widespread distribution of ideas.

It is apparent from the legislative history of the 1980 Amendent that the Commission believed that copyright protection of computer software is a function of economic imbalances. Careful attention to the Final Report will insure that interpretations of the 1980 Amendment are consonant with its economic underpinnings. Courts which ignore the legislative history embodied in the Commission's reports and rely solely upon the statutory language risk providing protection where it is not economically necessary and denying protection where the market-place demands it.