Fordham Law Review

1996

Interfaces and Interoperability in Lotus v. Borland: A Market-Oriented Approach to the Fair Use Doctrine

David R. Owen

Recommended Citation

INTERFACES AND INTEROPERABILITY IN *LOTUS V. BORLAND*: A MARKET-ORIENTED APPROACH TO THE FAIR USE DOCTRINE

David R. Owen

INTRODUCTION

The evolution of computer technology has strained the ability of copyright law to adapt to the modern world. Ever since courts first held that computer programs were "works of authorship"[1] subject to the protection of copyright law, authors and courts have been struggling to fit the new technologies into the copyright paradigm. Because of the continued expansion of computer technology and the growing importance of the computer industry to the U.S. economy, this unsettled area of law is more important than ever.[2]

Historically, copyright law has been quite flexible in adapting to new technologies and forms of creative expression. Copyright law has successfully accommodated new technologies such as photography, sound recording, and videotape.[3] The advent of computer technology has similarly resulted in the adaptation of many established principles of copyright law to an entirely new environment. As computer technology advances and its importance to the U.S. economy grows, copyright law must continue to serve its proper role of promoting "the Progress of Science and the useful Arts."[4]

Although the extension of copyright protection to computer programs was both natural and inevitable,[5] the area of law continues to be controversial and unsettled.[6] Certain characteristics of computer programs present unique problems for copyright and raise difficult issues for legal commentators and decision makers to recognize and resolve.[7] For example, can the manufacturer of a word processing program prohibit other manufacturers from selling a program that can access documents created by its program? Identification and resolution of such issues is necessary to advance the policy that copyright law is designed to address.

To function properly, computer technology depends on a framework for communication between computer programs and the com-

---

1. *See, e.g.*, Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171, 173 (N.D. Cal. 1981) (holding that a computer program is a 'work of authorship' ").
2. *See* Anthony L. Clapes, Software, Copyright, and Competition 19-20 (1989).
3. *See* Dan Rosen, *A Common Law for the Ages of Intellectual Property*, 38 U. Miami L. Rev. 769, 770-73 & 776-78 (1984).
4. U.S. Const. art. I, § 8, cl. 8.
5. Raymond T. Nimmer & Patricia A. Krauthaus, *Software Copyright: Sliding Scales and Abstracted Expression*, 32 Hous. L. Rev. 317, 328-29 (1995).
6. *Id.* at 318.
7. *See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308, 2315-16 (1994) [hereinafter *Manifesto*].

puter itself. Commentators have described this requirement as a need for "interoperability" between computer technologies.[8] Computer programmers must address technical and practical requirements relating to interoperability to design their programs effectively. A broad scope of copyright protection has the potential to exclude programmers from using material that is essential to interoperability.[9] Established market leaders could use copyright law to preclude competitive access to certain markets in any form. Conversely, a narrow scope of copyright protection may provide inadequate incentives to create new and useful programs.[10]

Interoperability also has significance for the computer user. To operate a computer program effectively, a computer user must spend a significant amount of time learning how the program works.[11] To assist the user, some software companies have attempted to provide programs that work in the same way as popular preexisting programs. Such conduct has raised a key copyright question: whether the law prohibits the copying of certain parts of a program to allow a computer user to employ the same skills to operate competing programs.[12]

The courts are currently in confusion concerning how copyright law should address such interoperability issues in software[13] copyright litigation.[14] *Lotus Development Corp. v. Borland International*,[15] a recent case that addressed interoperability issues, resulted in a 4-4 deadlock in the Supreme Court,[16] indicating both the importance of

---

8. " 'Interoperability' refers to the ability of consumers to use the products of different manufacturers interchangeably or in combination." Peter A. Wald et al., *Standards for Interoperability and the Copyright Protection of Computer Programs, in* Intellectual Property Antitrust: 1995, at 859 (PLI Patents, Copyrights, Trademarks, & Literary Property Course Handbook Series No. G4-3942, 1995).

9. *Id.*

10. *Id.*

11. *Manifesto, supra* note 7, at 2374-75.

12. *See* Linda Skon, Comment, *Copyright Protection of Computer User Interfaces: "Creative Ferment" In the Courts*, 27 Ariz. St. L.J. 1063, 1063-64 (1995) (noting that two courts have reached opposite conclusions regarding copyright protection for user interfaces).

13. Computer "software" is generally associated with the programs that allow computers to function, as opposed to the actual machine components which comprise the computer "hardware." *See Manifesto, supra* note 7, at 2319. There is some ambiguity and overlap between the two terms. *See infra* notes 128-31 and accompanying text.

14. *Compare* Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995) (holding that menu component of computer program's user interface is unprotectable under copyright law), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996) *with* Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994) (holding that input formats of computer program's user interface is protected expression under the copyright law).

15. 116 S. Ct. 804 (1996).

16. *Id.* at 805. Justice Stevens took no part in the case, and the remaining eight Justices could not agree on a majority opinion. *Id.*

the issue as well as the lack of consensus.[17] The Court, by default, affirmed the appellate court ruling that the copyright law did not protect the system of menus and commands used in Lotus' spreadsheet program 1-2-3. The deadlock was a major disappointment to lawyers, judges, and commentators who had expected the Court to resolve the confusion.[18]

The trial court and court of appeals in the *Lotus* case took opposite positions. The trial court focused on the importance of copyright protection for creative expression,[19] whereas the court of appeals focused on the importance of providing consumers with access to various programs through interoperability.[20] While both opinions addressed important features of copyright law, neither reached a satisfactory result. Instead the courts took positions that failed to effectively balance the important competing interests at stake. The failure of both courts to resolve the case properly demonstrates the need for a more complete analysis of interoperability issues in copyright law.

This Note argues that a proper analysis of interoperability issues is critical to the continuing viability of copyright's application to computer software. It presents a method of evaluating interoperability under the "fair use doctrine," which is an affirmative defense to a claim of copyright infringement. This Note asserts that a proper evaluation of interoperability should identify the protected markets served by the copyrighted work and the effect on those markets resulting from the contested use. Using an approach that focuses on the markets for computer software can provide for a proper evaluation of whether a particular use of copyrighted material is a "fair use" or will undermine the demand for the original. Where the market for a copyrighted work is unfairly harmed by a particular use, copyright must prohibit the use if it is to provide meaningful protection.

Part I introduces general principles and policies of copyright law and the fair use doctrine. Part II discusses the expansion of copyright law to cover computer programs, addressing the special requirements of computer technology in this area. Part III evaluates the role of interoperability in determining the scope of copyright protection for computer programs. Part IV discusses the facts of *Lotus* and contrast the district and circuit court opinions. Part V then evaluates the protection asserted by Lotus under current caselaw and applies the fair use doctrine in a way that takes into account interoperability issues. Part VI suggests some avenues for resolving the interoperability issues

---

17. Linda Greenhouse, *Supreme Court Deadlocks In Key Case on Software*, N.Y. Times, Jan. 17, 1996, at D2.

18. *Id.*

19. Lotus Dev. Corp. v. Borland Int'l, 799 F. Supp. 203, 223 (D. Mass 1992), *rev'd*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

20. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 817-18 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

raised by the *Lotus* case in a manner that is consistent with copyright policy. Finally, this Note concludes that the use of protected elements of computer programs that are necessary for interoperability should be allowed under copyright law subject to a requirement that unfair harm to the market for the "copyrighted work"[21] is minimized or eliminated.

I. COPYRIGHT EXISTS TO PROTECT PUBLIC ACCESS TO CREATIVE WORKS

The authority of copyright law to protect creative expression derives directly from the Constitution and aims to promote the advancement of the arts and sciences.[22] Copyright law advances this policy by granting a temporary monopoly on an author's "expression"[23] in order to reward the creative effort. The purpose of the copyright monopoly, however, is not specifically to reward authors, but instead to benefit the public by providing authors with an incentive to create.[24]

Most fundamentally, copyright protection precludes the unauthorized copying of the protected work.[25] It also gives the author an exclusive right to prepare "derivative works."[26] A "derivative work" is defined as one that is based on or developed from a preexisting work or works.[27] These rights vest upon the moment of creation.[28]

A claim under the copyright law depends on proof of two elements.[29] First, the plaintiff must show the ownership of a valid copy-

---

21. Although the works of both plaintiff and defendant in a copyright lawsuit are typically protected by copyright, for purposes of this Note, the "copyrighted work" will refer to the work of the party claiming infringement.

22. U.S. Const. art. I, § 8, cl. 8 ("Congress shall have Power . . . [t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.").

23. The types of expression expressly covered by the copyright statutes include: literary works, musical works, motion pictures, sound recordings, dramatic works, choreographic works, sculptural works, and architectural works. 17 U.S.C. § 102(a) (1994). The monopoly granted by copyright reserves for the copyright holder primarily the rights of reproduction (copying) and the preparation of derivative works. *Id.* § 106.

24. *See* Mazer v. Stein, 347 U.S. 201, 219 (1954); 1 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 1.03(A) (1995) [hereinafter Nimmer on Copyright] (stating that the primary purpose of copyright protection is to secure the public benefits of creative expression); *see generally* L. Ray Patterson & Stanley W. Lindberg, The Nature of Copyright: A Law of Users' Rights 2 (1991) (describing the role and purpose of copyright protection).

25. 17 U.S.C. § 106(1) (1994).

26. *Id.* § 106(2).

27. *Id.* § 101 (defining derivative work as "a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version . . . which, as a whole, represent an original work of authorship").

28. Rosen, *supra* note 3, at 775 (citing 17 U.S.C. §§ 102, 302(a) (1982)).

29. Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 361 (1991).

right.[30] Typically a plaintiff accomplishes this by producing a certificate of copyright registration.[31] Second, the plaintiff must show copying by the defendant.[32] If no direct evidence of copying is present, the copyright owner can alternatively prove that the defendant had access to the copyrighted work and that the two works are substantially similar.[33]

A. *Limits on Copyright Protection*

In order to keep the scope of the copyright monopoly reasonable, several principles have evolved to deny copyright protection where extending it would undermine the policy of benefitting the public. For example, copyright protects the author's expression, but not any ideas contained in that expression.[34] This distinction protects the right of the public to build upon and use any ideas that are incorporated into copyrighted expression.[35]

The distinction between what is an "expression" and what is an "idea" is one that has troubled the courts for a long time. Over 100 years ago in the seminal case of *Baker v. Selden*,[36] the Supreme Court held that the copyright of a book describing a double entry accounting system did not cover the "idea" of the system described in the book.[37] Thus, the public was free to use the accounting system described in the book, and the copyright protection granted to the book only served to prohibit the copying of the book itself.[38]

Similar restrictions are found in the section of the copyright law that precludes copyright protection for any "procedure, process, system, [or] method of operation . . . regardless of the form in which it is described."[39] For example, if a physician were to publish a book describing the method of preparing a certain drug, the copyright of the book would provide no exclusive right to manufacture or sell the

---

30. *Id.*

31. Although registration is not required for protection, it provides substantive and procedural advantages such as statutory damages and attorney's fees. John B. Koegel, *Bamboozlement: The Repeal of Copyright Registration Incentives*, 13 Cardozo Arts & Ent. L.J. 529, 531 (1995); I. Fred Koenigsberg, *Copyrights, in* Understanding Basic Copyright Law: 1995, at 22-23 (PLI Patents, Copyright, Trademark, & Literary Prop. Course Handbook Series No. G4-3946, July 10, 1995). Registration is accomplished by completing a form, paying a fee, and submitting deposit copies of the work with the U.S. Copyright Office, a branch of the Library of Congress. Koegel, *supra*, at 531-32.

32. *Feist*, 499 U.S. at 361.

33. Marvin N. Benn & Richard J. Superfine, *§ 117—The Right To Adapt Into the Fourth Generation and The Source Code Generator's Dilemma*, 12 J. Marshall J. Computer & Info. L. 537, 550 (1994).

34. Baker v. Selden, 101 U.S. 99, 101-02 (1879).

35. *See id.*

36. 101 U.S. 99 (1879).

37. *Id.* at 104.

38. *Id.*

39. 17 U.S.C. § 102(b) (1994).

drug.[40] These restrictions are consistent with copyright policy. For copyright law to benefit the public, the public must remain free to use the facts, ideas, and procedures that are embodied in a protected work.[41]

Other restrictions on copyright protection focus on the practical problems that can arise from a grant of copyright protection. The "merger doctrine" denies copyright protection where only a few ways exist to express a particular idea.[42] The theory is that because of the limited ways of expressing the idea, the expression and the idea have "merged," thus making a grant of copyright equivalent to a monopoly on the idea.[43] Using this reasoning, the Ninth Circuit denied copyright protection to the designer of a jeweled pin in the shape of a bee.[44] Due to the limited number of ways to design such a pin, the court found the idea and expression to have merged.[45]

Similarly, the *scènes-à-faire* doctrine withholds protection for elements that are customary or standard expressions in the treatment of a given idea or topic.[46] For example, the copyright of a novel about a particular historical era would not provide protection for descriptions or features that were typical of that era.[47]

B. *Fair Use As an Affirmative Defense*

In addition to restrictions on the scope of protection granted by copyright, the doctrine of "fair use"[48] provides an affirmative defense to the use of copyrighted material for certain purposes. The courts created the doctrine in the mid-1800s to permit limited infringement under the law, where the infringing work "serve[s] a specific public interest in the progress of art or knowledge through transformative creation of a new work."[49] The fair use doctrine provides a defense to what would otherwise be prohibited, infringing conduct.[50]

---

40. *See Baker*, 101 U.S. at 102-03.
41. *See* Nimmer & Krauthaus, *supra* note 5, at 323.
42. Jon S. Wilkins, Note, *Protecting Computer Programs as Compilations Under Computer Associates v. Altai*, 104 Yale L.J. 435, 438-39 (1994).
43. *Id.*
44. Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971) (concluding that "[t]here is no greater similarity between the pins of plaintiff and defendants than is inevitable from the use of jewel-encrusted bee forms in both").
45. *Id.*
46. Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 709 (2d Cir. 1992).
47. *See* Hoehling v. Universal City Studios, Inc., 618 F.2d 972, 979 (2d Cir. 1980).
48. 17 U.S.C. § 107 (1994).
49. Karen S. Frank & Michael J. Higgins, *Fair Use: In the Courts and Out of Control?*, *in* Advanced Seminar on Copyright Law: 1995, at 5-6 (PLI Patents, Copyright, Trademark, & Literary Prop. Course Handbook Series No. G-411, 1995) [hereinafter Frank & Higgins] (citing Folsom v. Marsh, 9 F. Cas. 342 (C.C.D. Mass. 1841) (No. 4,901)). In fact, Justice Story held in *Folsom* that the defendant's work was not sufficiently in the public interest to excuse the infringement, but the analysis used provided the basis for the development of the fair use doctrine. *Id.* at 6-7.
50. *Id.* at 4.

The fair use doctrine was codified at 17 U.S.C. § 107 as part of the Copyright Act of 1976.[51] The statutory recognition of fair use was intended to " 'restate the present *judicial* doctrine of fair use, not to change, narrow, or enlarge it in any way.' "[52] The statute lists four factors that should be balanced in determining whether a challenged use is "fair":

> (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
> (2) the nature of the copyrighted work;
> (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
> (4) the effect of the use upon the potential market for or value of the copyrighted work.[53]

If the disputed use is fair in light of these criteria, the defendant is not liable for infringement.[54]

The fair use doctrine only applies after a finding that the plaintiff's material is protected ("copyrightable") and that the defendant has copied the plaintiff's expression ("infringement").[55] Thus, in the evaluation of a fair use defense, the focus shifts to the character and effect of the defendant's use and away from the defendant's alleged copying.

Fair use analysis is an important component of copyright law. It permits the application of copyright policy to cases where the fact of copying is not in dispute, but the character of the contested use is one that arguably furthers rather than undermines the goals of copyright.[56] By preventing certain illogical outcomes, the fair use doctrine enhances the legitimacy and durability of the copyright statutes.

The early 1990s have seen an increase in the number of cases addressing the fair use doctrine. Historically, the doctrine was not available where the otherwise infringing use served a commercial purpose. In *Sony Corp. v. Universal City Studios, Inc.,*[57] the Supreme Court stated that "every commercial use of copyrighted material is presump-

---

51. Leo J. Raskind, *The Uncertain Case for Special Legislation Protecting Computer Software*, 47 U. Pitt. L. Rev. 1131, 1175 (1986).

52. William F. Patry, The Fair Use Privilege in Copyright Law 413 (2d ed.1995) (quoting H.R. Rep. No. 1476, 94th Cong. 2d Sess. 66 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5680).

53. 17 U.S.C. § 107 (1994). Evaluation of a fair use defense is not necessarily limited to consideration of these four factors, and on occasion courts have considered other relevant factors, such as the defendant's good faith. Patry, *supra* note 52, at 415 & n.14.

54. *See* 17 U.S.C. § 107.

55. Patry, *supra* note 52, at 413.

56. *Cf.* Raskind, *supra* note 51, at 1175 ("[U]se of protected material for purposes of literary criticism and commentary, for news reporting, for classroom use and teaching, and for scholarship and research constitutes a fair use.").

57. 464 U.S. 417 (1984).

tively ... unfair."[58] The presumption discussed by the *Sony* Court was limited, if not eliminated, by the more recent fair use decision in *Campbell v. Acuff-Rose Music, Inc.*,[59] which stated: "If, indeed, commerciality carried presumptive force against a finding of fairness, the presumption would swallow nearly all of [the rule]."[60] This shift in the presumption against commercial uses has expanded the applicability of the fair use defense, particularly in the area of consumer computer software.

II. APPLICATION OF COPYRIGHT TO COMPUTER PROGRAMS

As new forms of communication technology develop, pressure is put on the copyright law to adapt in a way that provides reasonable protection for the new forms of expression. Modern copyright law has evolved from Constitutional guidelines to cover motion pictures, record albums, and compact disks.[61] Computer technology is therefore not unique in this respect. Nevertheless, the protection of computer software under the copyright paradigm requires the courts to grapple with certain distinctive issues.

This part discusses the adaptation of copyright law to cover computer programs. Part II.A addresses the special characteristics of computer programs under copyright law. Part II.B will discuss recent cases that are shaping the scope and limitations of copyright protection in this area, and part II.C discusses the application of the fair use defense in the computer software context.

A. *Computer Programs: A Special Case*

Little question exists that computer software demands some degree of copyright protection.[62] Designing and writing computer software requires a high degree of creativity.[63] Computer programs are also generally designed to serve a particular purpose, and thus provide a degree of utility that is different from most works of art and literature. This utilitarian aspect distinguishes computer programs from other works protected by copyright law. Additionally, computer software must be able to operate in an environment that demands a high level of cooperation between the hardware on which it runs and the user who operates the software. As one court observed:

> The problem presented by computer programs is fundamentally different in one respect. The computer program is a means for causing something to happen; it has a mechanical utility, an instrumental

---

58. *Id.* at 451.
59. 114 S. Ct. 1164 (1994).
60. *Id.* at 1174.
61. National Research Council, Intellectual Property Issues in Software 12 (1991) [hereinafter NRC].
62. *See* Nimmer & Krauthaus, *supra* note 5, at 320.
63. *Id.* at 321.

> role, in accomplishing the world's work. . . . Utility does not bar copyright (dictionaries may be copyrighted), but it alters the calculus.[64]

This incongruity between computer programs and the type of protection traditionally conceived under copyright requires a specialized examination of the type and scope of protection needed for computer software.[65]

Although a great deal of creativity is incorporated into computer software,[66] functional and utilitarian considerations are also necessary for software to achieve its desired purpose.[67] Some commentators have seized upon this problem as demonstrating the need for an entirely new form of intellectual property protection applicable only to the software arena.[68] By properly analyzing the scope of copyright protection under existing principles of copyright law, however, courts can eliminate the need to develop an entirely new body of law.[69]

In many ways the extension of copyright protection to computer software was both necessary and inevitable.[70] Computer software is susceptible to easy and quick copying without any degradation between the copy and the original.[71] As some commentators observed:

> Copyright law, with its low threshold for protection and specific prohibition against copying, was a natural response. Patent and trade secret laws are inadequate because they require either maintained secrecy, which is inconsistent with mass markets, or an advanced degree of innovation, which is inconsistent with an evolutionary industry involving many products.[72]

Copyright thus supplies an existing, accessible body of law for the protection of creative investment in computer software.

The earliest litigation involving copyright protected computer programs primarily addressed the easier case of exact copying of entire programs.[73] In *Apple Computer, Inc. v. Franklin Computer Corp.*,[74] the Third Circuit found that copyright prohibited wholesale copying of a computer operating system program whether in "source" or "ob-

---

64. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 819 (1st Cir. 1995) (Boudin, J., concurring) (emphasis omitted), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

65. *See* NRC, *supra* note 61, at 12-13.

66. *See* Patry, *supra* note 52, at 519.

67. *See, e.g.*, Clapes, *supra* note 2, at 22-23 (observing that a computer program must be written to run on a particular type of computer).

68. *See, e.g.*, *Manifesto*, *supra* note 7, at 2364-65 (arguing in favor of the development of a new legal regime to protect software).

69. *See* Raskind, *supra* note 51, at 1182.

70. Nimmer & Krauthaus, *supra* note 5, at 328.

71. *Id.* at 328-29.

72. *Id.* at 329.

73. *Id.*

74. 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

ject" code format.[75] Such cases are easier to evaluate because they involve a direct theft of expression that fits well within the traditional prohibitions of copyright law.

More recently, courts have been forced to examine the more difficult issues that arise in cases where the defendant's copying has been (1) limited to specific program elements, or (2) directed at achieving functionality or interoperability of computer products. The former set of cases have focused primarily on distinguishing the protected and the unprotected elements of plaintiff's work, a "copyrightability" question.[76] The second set of cases focus on the legitimacy of the defendant's conduct in copying the plaintiff's work,[77] a "fair use" question. Although the two categories overlap to some extent,[78] they are different enough to warrant separate treatment.

B. *Copyrightability*—Whelan *to* Altai

The "literal" elements of computer programs include primarily the "source"[79] code and the translated "object" code.[80] These components indisputably enjoy copyright protection.[81] Copying "nonliteral" elements involves the copying of particular features, options, or appearances of another program. Copyright protection also extends to some nonliteral elements; otherwise immaterial variations from the original would allow a plagiarist to escape liability.[82] Nonliteral copying of computer software has been more difficult to evaluate under traditional principles of copyright law.[83]

The trend in the early 1980s was to provide expansive protection for a program's nonliteral elements.[84] In *Whelan Associates v. Jaslow Dental Laboratory, Inc.*,[85] a case that some commentators consider to

---

75. Id. at 1249. "Source code" indicates the human readable instructions actually typed in by the programmer. "Object code" refers to the machine readable instructions that tell the computer how to behave. *Id.* at 1248. During a process called "compilation," the source code is translated into object code so that the computer can understand and perform the program instructions. *See* Clapes, *supra* note 2, at 63-64.

76. *See* Nimmer & Krauthaus, *supra* note 5, at 318 (using the terms "evidentiary" and "liability" instead of "copyrightability" and "fair use").

77. *Id.*

78. For example, a court seeking to find for the defendant in a particular case may accomplish that result by holding that the plaintiff's work is not protected, or alternatively that infringement by the defendant is a fair use.

79. For a definition of "source" and "object" code, see *supra* note 75.

80. Wald, *supra* note 8, at 868.

81. *Id.*

82. The fear is that by inserting minor and unimportant differences in a copied work, the copying will not be prohibited by copyright law. *See* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930).

83. Wald, *supra* note 8, at 875.

84. Pamela Samuelson, *Counterpoint: An Entirely New Legal Regime is Needed*, 12 Computer Law. 11, 11 (1995) [hereinafter Samuelson, *Counterpoint*].

85. 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

be the "high-water mark" of copyright protection,[86] the Third Circuit held that the general purpose or function of a computer program constituted the program's "idea."[87] The court then reasoned that everything else "necessary to effecting that function" would constitute protected expression.[88] The result was a rule that found only one unprotected idea in each program.[89]

The one idea per program principle described in *Whelan* came under heavy criticism as providing too much protection for a program's nonliteral elements.[90] Critics argued that "[t]he crucial flaw in [the *Whelan*] reasoning is that it assumes only one 'idea,' in copyright law terms. . . . The broad purpose that a programs serves . . . is an idea. Other elements of the program structure and design, however, may also constitute ideas for copyright purposes."[91]

Courts responded to these criticisms by narrowing the scope of protection.[92] In *Computer Associates International, Inc. v. Altai, Inc.*,[93] the Second Circuit set forth what is considered "the leading test of software copyright infringement."[94] One commentator observed that "although the Third Circuit . . . has yet to comment on *Altai*, every other court to consider the issue has endorsed at least some aspects of the *Altai* approach."[95] The *Altai* test consists of three steps described as "abstraction", "filtration", and "comparison."[96] Under *Altai*, the copyrightability question is determined by the first two steps of abstraction and filtration, both of which strip the plaintiff's work of elements unprotected by copyright.[97] Whatever remains after the first two steps is then compared to the allegedly infringing work to determine liability.[98]

---

86. Wald, *supra* note 8, at 875.

87. *Whelan*, 797 F.2d at 1236.

88. *Id.*

89. *See* Wilkins, *supra* note 42, at 442 n.36.

90. *See* 3 Nimmer on Copyright, *supra* note 24, § 13.03[F] at 13-126-28, 131-33; Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan. L. Rev. 1045, 1074 (1989).

91. 3 Nimmer on Copyright, § 13.03[F], at 13-131-32.

92. Wald, *supra* note 8, at 876-78.

93. 982 F.2d 693 (2d Cir. 1992).

94. Wilkins, *supra* note 42, at 435.

95. *See* Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 S. Cal. L. Rev. 1091, 1141 (1995). Courts adopting the Second Circuit *Altai* test include the Fifth Circuit in Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1342-43 (5th Cir. 1994), the Tenth Circuit in Autoskill Inc. v. National Educational Systems, Inc., 994 F.2d 1476, 1490-91 (10th Cir. 1993), and the Ninth Circuit in Sega Enterprises Ltd. v. Accolade Inc., 977 F.2d 1510, 1524 (9th Cir. 1992).

96. Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 706-12 (2d Cir. 1992).

97. *See* Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977, 1002 (1993).

98. *Id.* at 1002-03.

"Abstraction" determines the scope of protection that copyright affords to particular program elements by separating the unprotected ideas from the protected expression.[99] It determines where along the spectrum of ideas and expressions a particular program element lies.[100] The "filtration" step then filters out any elements that are unprotected for reasons of merger, scènes-à-faire, and the like.[101] Whatever is left following the first two steps is therefore protected expression. The final "comparison" step compares the protected portions to the challenged work to see if there is a substantial similarity between the two.[102]

One drawback of the *Altai* approach is that courts may find it difficult to apply consistently.[103] Many of the issues addressed in the three stages involve complex analyses on a case-by-case basis.[104] Some commentators also criticize the method as providing insufficient protection for valuable elements of computer software.[105] The superiority of the *Altai* approach, however, lies in its provision of the necessary sophistication to identify and address the difficult questions raised by the copyrightability of nonliteral elements of computer programs.

C. *Fair Use*

Another area of copyright law that has recently developed in the context of computer programs is the fair use doctrine. Fair use is " 'a privilege in others than the owner of the copyright to use the copyrighted material in a reasonable manner without [the owner's] consent.' "[106] In essence, fair use is excusable copyright infringement.

A number of circuit court opinions have expanded the scope of the fair use doctrine to accommodate new issues raised by cases involving copyrighted computer game programs. One category of these cases has addressed the process of "reverse engineering."[107] Reverse engi-

---

99. *Altai*, 982 F.2d at 706-07.
100. *See id.*
101. *Id.* at 707-10.
102. *Id.* at 710-11.
103. *See, e.g.,* Bateman v. Mnemonics, Inc., No. 93-3234, 1995 WL 757786 at *5-7 (11th Cir. Dec. 22, 1995) (holding that the district court failed to properly instruct the jury on the applicability and scope of the *Altai* test); Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1343 n. 10 (5th Cir. 1994) (noting the difficulty of applying the abstraction method).
104. *See, e.g.,* Bateman, 1995 WL 757786, at *8 (criticizing the district court for failing to adequately define terms in the jury instructions that were necessary for resolving the case).
105. Miller, *supra* note 97, at 1004-05 (arguing that the *Altai* test has the potential to improperly filter out the most expressive components of a computer program where a programmer has captured the most effective way of communicating an idea).
106. Frank & Higgins, *supra* note 49, at 4 (quoting Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 549 (1985)).
107. Reverse Engineering involves the translation of machine readable "object" code back into human readable "source" code (a process known as "decompiling")

neering describes the process of figuring out the internal operation of a computer program by examining, decompiling, and using the program.[108] A software developer often needs to reverse engineer another designer's program in order to design a product that is technically compatible with it.

In *Sega Enterprises Ltd. v. Accolade, Inc.*,[109] a case that addressed reverse engineering, the Ninth Circuit held that the fair use doctrine allowed the defendant Accolade to make unauthorized copies of plaintiff's programs in order to figure out how to make its own game cartridges compatible with the plaintiff's Genesis home video game system.[110] The *Sega* court noted the importance of considering "the public benefit resulting from a particular use notwithstanding the fact that the alleged infringer may gain commercially."[111] Thus, the process of reverse engineering was sanctioned by the court for limited purposes aimed at achieving interoperability.[112]

The *Sega* court based its holding primarily on two conclusions. First, the court decided that the public benefitted from copying to facilitate interoperability between Accolade's games and the Genesis system.[113] Second, the court concluded that Accolade's copying would result in no direct harm to the market for Sega's games.[114] On this point the court stated:

> Accolade's [copying] has led to an increase in the number of independently designed video game programs offered for use with the Genesis console. It is precisely this growth in creative expression . . . that the Copyright Act was intended to promote.
>
> . . . .
>
> [V]ideo game users typically purchase more than one game. . . . In any event, an attempt to monopolize the market by making it impossible for others to compete runs counter to the statutory purpose of promoting creative expression . . . .[115]

---

for the purposes of gaining a better understanding of the internal operation of a computer program. Patry, *supra* note 52, at 468. It necessarily requires the making of an unauthorized "copy" of the program. *See* Sega Enters. Ltd. v. Accolade Inc., 977 F.2d 1510, 1518-19 (9th Cir. 1992).

108. Patry, *supra* note 52, at 468.

109. 977 F.2d 1510 (9th Cir. 1992).

110. *Id.* at 1527-28. Around the same time, the Federal Circuit adopted the same reasoning and stated that reverse engineering from a legitimately obtained copy is a fair use. Atari Games Corp. v. Nintendo of Am., 975 F.2d 832, 843 (Fed. Cir. 1992).

111. *Sega*, 977 F.2d at 1523.

112. *Id.* at 1527-28.

113. *Id.* at 1527.

114. *Id.* at 1523-24.

115. *Id.* The *Sega* ruling was cited approvingly by the Supreme Court in Campbell v. Acuff-Rose Music, Inc., 114 S. Ct 1164, 1174 (1994).

Thus, the court rejected the argument that the use of copyrighted material to achieve interoperability and produce a competing product could never be a fair use.[116]

Another case that expanded the scope of the fair use doctrine in the context of computer software is *Lewis Galoob Toys v. Nintendo of America*.[117] Galoob manufactured a device called the "Game Genie" which the user interposed between the game cartridge and the video game console for the purpose of changing the performance of the games. By using the Game Genie, a player could, among other things, increase the number of lives of a particular video game character.[118]

Nintendo sued Galoob alleging contributory copyright infringement. Nintendo asserted that Galoob was selling the Game Genie with the knowledge and intention that consumers would use it to modify the behavior of Nintendo's games.[119] Under Nintendo's theory, the modified games were unauthorized "derivative works"[120] based on the original game programs. Although the Court of Appeals doubted the merit of Nintendo's derivative work theory,[121] it held that even assuming that the changes made to the game did amount to an unauthorized derivative work, any infringement arising thereby would be a fair use.[122]

The relevance of the *Galoob* reasoning to interoperability issues should not be missed. Like the Game Genie, many computer programs interact with preexisting works.[123] For example, a spell-check program must interact with a word processing program. A word processor document must interact with word processing programs. Such interaction may at times implicate the copyright laws. Under *Galoob*, the most important fair use evaluation appears to be the effect on the market for the preexisting work.[124]

As these cases point out, technological issues that are unique to computer programs require that copyright analysis include accommodation for certain uses of copyrighted material. This circumstance demands that the fair use doctrine be flexible enough to adapt to this new environment. As one commentator has noted: "Fair use is an

---

116. Patry, *supra* note 52, at 469-70.
117. 964 F.2d 965 (9th Cir. 1992), *cert. denied*, 113 S. Ct. 1582 (1993).
118. Lewis Galoob Toys v. Nintendo of Am., 780 F. Supp. 1283, 1289 n.2 (N.D.Cal. 1991).
119. *See id.* at 1286-89.
120. A "derivative work" is a work that is based at least in part on another work. 17 U.S.C. § 101 (1994). The right to create a derivative work requires authorization from the original copyright holder. *Id.* § 106 (2); *see supra* notes 26-27 and accompanying text.
121. *Galoob*, 964 F.2d at 969.
122. *Id.* at 969-72.
123. *See* Pamela Samuelson, *Fair Use for Computer Programs and Other Copyrightable Works in Digital Form: The Implications of Sony, Galoob and Sega*, 1 J. Intell. Prop. L. 49, 102, 112 (1993) [hereinafter Samuelson, *Fair Use*].
124. *See id.* at 78.

'equitable rule of reason' of *general* application for mediating between the interests of consumers and of copyright owners as to uses that can be made of the protected work."[125] Courts must adapt this principle to the context of interoperability of computer software in a way that gives effect to the basic principles underlying copyright.

III. INTEROPERABILITY AND ITS ROLE IN COPYRIGHT PROTECTION FOR COMPUTER PROGRAMS

Computer programs are not like other copyrighted literary works, such as books, because computer programs must work together with other programs and machines to bring about the desired result.[126] A critically important task of software development is to design information flow in a manner that allows the user to achieve the objective.[127] If, for example, a program is written so that it cannot interoperate with the computer on which it runs, its usefulness is frustrated. Where the various components that allow for interoperation may be subject to copyright protection, the potential threat is that copyright will inhibit the advancement of science and the arts by preventing software designers from achieving interoperability. This section describes different types of interoperability. It will then evaluate the impact that interoperability issues can have on the software markets through the operation of copyright law.

A. *Different Types of Interoperability*

Interoperability requirements take many different forms. The most obvious of these is the ability of a particular program (software) to run on a particular machine (hardware).[128] This ability can be understood as "hardware-software" interoperability. Similarly, some programs must be able to interoperate with other programs.[129] For a spell-checking program to be useful, it must be able to work together with a word processor.[130] This ability could be similarly understood as "software-software" interoperability.

Emphasis on the hardware-software distinction can be misleading, however, because "[a]ny behavior that can be accomplished with one can also be accomplished with the other."[131] Video game cartridges provide a good example. Although in some sense the cartridges are

---

125. *Id.* at 99 (emphasis added); *see also* Raskind, *supra* note 51, at 1182 ("[I]t is possible that the fair use doctrine could be adapted to serve as an arbiter between permissible use of protected software consistent with the public interest in technological progress and impermissible piracy.").

126. *Manifesto*, *supra* note 7, at 2320-21.

127. *Id.* at 2321.

128. *See* Clapes, *supra* note 2, at 22-23, 174.

129. *Id.* at 174-75.

130. *See* Lewis Galoob Toys, Inc. v. Nintendo of Am., 964 F.2d 965, 969 (9th Cir. 1992), *cert. denied*, 113 S. Ct. 1582 (1993).

131. *Manifesto*, *supra* note 7, at 2319.

"hardware," they also constitute the type of computer program that is ordinarily associated with "software."[132] To the video game user, of course, the distinction is irrelevant.

Other forms of interoperability go beyond mere technical compatibility. The ability of a user of a particular program to employ the same skills and experience to operate a different but similar program might be understood as "user-software" interoperability.[133]

One example of user-software interoperability occurs when programs accept documents created by the user's operation of other programs. If an author can access and manipulate the same document in a number of different word processing programs, then the word processors are interoperable with the author's document. Conversely, a word processing program that cannot access the author's document is not interoperable with it.[134]

This interoperability principle also applies to other forms of user created documents such as spreadsheet documents or software "macros." A "macro" is a computer program that is designed to shorten repetitive tasks that are frequently performed.[135] For example, a program user might create a macro that caused the word processor to translate all footnotes in a particular document into endnotes. The ability to use the macro on a variety of different word processors could have significant value to the user.[136]

Another example of user-software interoperability involves the ability of a certain program to "emulate" or behave similarly to another program so that the user does not need to learn a new set of skills when using the second program.[137] Programs that have similar

---

132. For a brief description of hardware and software, see *supra* note 13.

133. *See* Clapes, *supra* note 2, at 175-76 (discussing the "interaction between a program and its end user").

134. The ability to access different kinds of documents usually takes two forms. Either the document is translated into the "native" format of the program for manipulation (translation), or the program interprets any differences between the actual document format and the program's native format (interpretation). There is some authority that copyright should treat such approaches differently. *See, e.g.*, Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 69 (D. Mass. 1990) (distinguishing Microsoft's "Macro Translation Assistant" from Paperback's macro interpreter). It seems illogical to argue that copyright law distinguishes between these approaches because of the number of steps employed to achieve the same result.

135. Clapes, *supra* note 2, at 72. A "macro" is a kind of computer program written by the program user which speeds up the performance of routine or repetitive tasks by recording a series of menu actions and playing them back upon command. *Paperback*, 740 F. Supp. at 64.

136. *See, e.g.*, Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 809-10 (1st Cir. 1995) (discussing the benefits of macros in the context of computer spreadsheets), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

137. *See, e.g.*, *Lotus*, 49 F.3d at 810 (discussing the "Lotus Emulation Interface" that allowed the users of one spreadsheet program to apply their knowledge to another spreadsheet program).

"user interfaces"[138] attempt to capitalize on this learning by providing a level of familiarity to the user.[139] Once a computer user has dedicated a significant amount of time to learning a particular program, for the user to operate a similar but differently conceived program is wasteful and inefficient.[140]

For copyright to provide the proper scope of protection for computer programs, courts must properly address these interoperability issues. If copyright fails to account properly for interoperability, the markets for computer programs will suffer from failures that arise from overprotection and underprotection of copyrighted works.[141]

B. *Overprotection and Underprotection*

The goal of copyright is to advance the progress of science and the arts.[142] Copyright serves this goal by protecting creative expression against unauthorized copying. If creative works received no protection, the incentive to create would be inadequate, and the amount of creative activity would decline.[143] By contrast, an overbroad scope of copyright protection including, for example, the protection of ideas, would stifle the creative process and preclude people from building upon the creations of others.[144]

This general principle of intellectual property law is especially true in the world of computer programs. If too little protection is granted for computer programs, the incentive to develop new products is drastically reduced because the reward is gone.[145] The effect would be magnified in the computer context due to the ease with which computer programs can be copied.[146] If copyright protected only the programming code, a successful program might quickly be driven out of

---

138. The term "user interface" describes the way that the user interacts with a particular program. It includes the appearance of information on the screen, the method of inputing data, the descriptions of program operation, and the method of controlling the program behavior. Clapes, *supra* note 2, at 91-92.

139. *See id.*

140. *Lotus,* 49 F.3d at 817-18.

141. *See Manifesto, supra* note 7, at 2356-57.

142. U.S. Const. art. I, § 8, cl. 8.

143. Wendy J. Gordon, *Asymmetric Market Failure and the Prisoner's Dilemma in Intellectual Property,* 17 U. Dayton L. Rev. 853, 854 (1992).

144. *See* Wilkins, *supra* note 42, at 438-39.

145. To the extent that new products are developed, the motivation to hide or hold back the valuable information contained in a computer program by any technological means possible would be quite strong. As a result, the ability to build on the discoveries of others would also be reduced. *See, e.g.,* Miller, *supra* note 97, at 1005 & n. 132 (arguing that a narrow scope of protection can preclude copyright coverage for the most successfully expressive components of computer programs).

146. *See* Nimmer & Krauthaus, *supra* note 5, at 328-29.

the market by low cost "clones"[147] developed with "relatively trivial effort."[148]

Conversely, the overprotection of computer programs also runs a significant risk of causing market failures.[149] If one software manufacturer is granted a copyright monopoly in a way that precludes other competitors from entering the market in any form, the level of creative development would similarly decline. In addition, product designers would have an incentive to develop products that did not work with the products of other manufacturers in the hope of leveraging an existing monopoly into other markets, such as the markets for complementary hardware and software.[150] Eventually, the utility enjoyed by the public would significantly decline.

The health of the computer industry depends on maintaining a level of copyright protection that both encourages and rewards development, but does not excessively restrict the development of new products that build upon or advance the creations of the past. Interoperability issues play a central role in this regard.[151]

C. *Interoperability, Fair Use, and Efficient Markets*

Copyright protection serves as a market-correcting mechanism that guarantees a reward for creative development.[152] When copyright is applied to a new form of expressive technology, the law must adapt in a way that continues to protect the markets for both new and existing expression. Courts should therefore evaluate interoperability issues from a market-oriented perspective.[153]

---

147. A "clone" is a program that duplicates all of the functionality of another program without duplicating any of the programming code. *See* Clapes, *supra* note 2, at 153-55.

148. *Manifesto, supra* note 7, at 2379. The effect of clones in the market can lead to an additional cost arising from the underprotection of computer software and the resulting lack of innovation. This cost arises where the consumers of computer programs become too attached to an outmoded "standard" that no longer represents the most efficient method of accomplishing a certain result. For example, the QWERTY typewriter keyboard was originally advantageous because it restrained the speed of typists to a rate that could be accommodated by the old mechanical typewriters. The lack of innovation in this area has resulted in a standard keyboard that is inefficiently serving modern needs. *See* Joseph Farrell, *Standardization and Intellectual Property*, 30 Jurimetrics J. 35, 37-38 (1989).

149. One commentator observed that the danger of overprotection is greater in the area of utilitarian works such as computer programs. Jonathan Band, Lotus v. Borland *Through the Lens of Interoperability*, 12 Computer Law. 1, 5 (June 1995).

150. Frederick R. Warren-Boulton et al., *Point: Copyright Protection of Software Can Make Economic Sense*, 12 Computer Law. 10, 19-20 (Feb. 1995).

151. Band, *supra* note 149, at 6-7.

152. *See* Manifesto, *supra* note 7, at 2369-70; *see also* Gordon, *supra* note 143, at 853-54 (presenting two different economic models that underlie intellectual property protection).

153. Some commentators who argue for a market-oriented approach to software protection also argue that only *sui generis* protection can adequately address the protection requirements of computer programs. The *Manifesto Concerning the Legal*

The fair use doctrine is a necessary element of this debate for a number of reasons.[154] Often, a court's evaluation does not end with a finding that plaintiff's expression is protected and defendant copied it, rather, the analysis begins at that point.[155] The fair use doctrine not only recognizes this fact, it is based on it.[156] The doctrine also expressly addresses the policies and market considerations that are at the heart of copyright law.[157]

The copying of nonliteral elements of copyrighted computer programs may have a number of different effects on the markets for the copyrighted work. In some cases, these effects may tend to undermine the market for the original, and therefore defeat the policy goals of copyright.[158] By contrast, other uses may further the goals of copyright by enhancing public benefit.[159] Thus, the primary issue in these interoperability cases should be the effect of the use, not the copying itself.[160]

This Note argues that interoperability uses may serve to increase public utility, and proposes a market-oriented approach to determine if such uses should be prohibited.[161] If a contested use facilitates interoperability and does not unfairly impact on the market for the copyrighted work, the use should be sanctioned under the fair use doctrine. Conversely, where the contested use directly harms the market for the copyrighted work, copyright law must prevent the use

---

Protection of Computer Programs, *supra* note 7, argues that the special characteristics of computer programs are "obscured by the issues that copyright doctrine seems to demand." *Id.* at 2429.

The need for or advisability of *sui generis* protection for computer programs has been questioned by several other authors who support both a market-oriented approach, and are confident that existing principles of copyright can be adapted to properly address the new issues raised by protecting computer programs under copyright. *See, e.g.*, Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright Over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev 2559 (1994) (arguing that both practical and substantive reasons mitigate against creating a *sui generis* form of protection for computer programs); Miller, *supra* note 97, at 1034-36 (arguing against the suitability of *sui generis* protection for computer software); Raskind, *supra* note 51, at 1182 (arguing in support of the fair use doctrine as a preferable substitute for *sui generis* protection).

154. *See* Raskind, *supra* note 51, at 1175.

155. *See, e.g.*, Lotus Dev. Corp v. Borland Int'l, 49 F.3d 807, 812 (1st Cir. 1995) (noting that the defendant did not dispute the fact of copying, but contended that such copying was lawful), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

156. *See* Patry, *supra* note 52, at 413 (noting that the fair use defense is "relevant only after there has been a prima facie showing of infringement").

157. 17 U.S.C § 107(1), (4) (1994).

158. *See Manifesto*, *supra* note 7, at 2379-80 (pointing out that where important program elements can be copied without restriction, the result is destruction of the markets for the original work).

159. Sega Enterprises Ltd. v. Accolade Inc., 977 F.2d 1510, 1523 (9th Cir. 1992) (holding that the use of copyrighted material supports a public policy).

160. *See* Raskind, *supra* note 51, at 1133 n.5.

161. *See infra* part V.

if the law is to provide any meaningful protection to computer software.

IV. THE *LOTUS* CASE FROM AN INTEROPERABILITY PERSPECTIVE

The case of *Lotus Development Corp. v. Borland International,*[162] which recently resulted in a 4-4 deadlock in the Supreme Court,[163] provides an excellent context in which to explore interoperability issues in copyright law and the fair use doctrine. Observers expected the case to be a tremendously important decision, one many thought would "determine the future of the software industry."[164] When Justice Stevens recused himself, however, the remaining eight justices could not reach an opinion.[165] As a result of the deadlock, the future of this area of law is uncertain. This section will discuss the facts of *Lotus,* and the district and circuit court opinions. It will then contrast the different considerations upon which each opinion was based.

Lotus Development Corp., creator of the popular spreadsheet 1-2-3, filed the case in 1990 against Borland International, creator of a competing spreadsheet program called Quattro Pro.[166] Borland introduced Quattro Pro in 1989, and the program enjoyed a positive reception by the software industry.[167] In the words of Gary Reback, Borland's counsel, "Quattro Pro . . . won every major award for spreadsheet excellence . . . [and] invariably ranked substantially higher than 1-2-3 in head-to-head reviews and user comparisons, including those conducted by Lotus."[168]

Lotus contested two features of Quattro Pro, both of which required incorporating copies of the 1-2-3 menu system into Quattro Pro. First, Quattro Pro included a feature called the "Lotus Emulation Interface,"[169] which allowed Quattro Pro users to employ a copy of the 1-2-3 menu in Quattro Pro. Quattro Pro's designers included this option as an alternative to Quattro Pro's own menu system. They intended the option to make Quattro Pro more attractive to existing 1-2-3 users who wanted the extra functionality but did not want to

---

162. 116 S. Ct. 804 (1996).
163. *Id.*
164. Victoria Slind-Flor, *High Court Spends Snowy Session Hearing IP Cases,* Nat'l L.J., Jan. 22, 1996, at B1 ("The stakes are astronomically high for the multibillion-dollar software industry, which desperately needs to know the nature and size of the legal obstacles to developing product compatibility . . . .")
165. *Lotus,* 116 S. Ct. at 804.
166. Linda Greenhouse, *Supreme Court Deadlocks In Key Case on Software,* N.Y. Times, Jan. 17, 1996, at D2.
167. Gary L. Reback, *Implications of* Lotus v. Borland, *in* Computer Software and the Internet, at 169 (PLI Patents, Copyrights, Trademarks, & Literary Prop. Course Handbook Series No. G4-3953, 1995).
168. *Id.*
169. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 810 (1st Cir. 1995), *aff'd by an equally divided Court,* 116 S. Ct. 804 (1996).

learn to use a new spreadsheet program.[170] Second, Lotus also con-
tested the Quattro Pro "Key Reader" feature, which allowed macros
written by 1-2-3 users to run on Quattro Pro to the same effect.[171]

The 1-2-3 menu system, like all computer program menus, allows
the spreadsheet user to access the functions that the 1-2-3 program
can perform by using a keyboard or a mouse. Specifically, the 1-2-3
menu system covers 469 commands, which are arranged into over fifty
menus and submenus.[172] Borland admitted using the 1-2-3 menu, but
contended that either the menu was outside the protection of copy-
right or that the use was fair.[173]

The district and circuit court opinions took opposing views of Bor-
land's use of the 1-2-3 menu. The district court concluded that the 1-
2-3 menu was properly within the scope of Lotus' copyright and that
Borland had infringed that copyright.[174] The circuit court reversed,
holding that the 1-2-3 menu was an unprotectable "method of opera-
tion."[175] Both opinions addressed important issues in copyright, but
neither was able to reconcile the opposing considerations. The follow-
ing subparts will discuss each opinion in detail.

A. *The District Court Opinion*

The district court evaluated the 1-2-3 menu system primarily from
the perspective of the menu emulation.[176] Because copyright does not
protect ideas, but only the expression of ideas,[177] the district court saw
the case as turning on the separation of the protected expressions
from the unprotected ideas.[178]

The district court held that because of the great variety of ways that
a programmer could set up a system of menu commands in a spread-
sheet program, the particular choices made by the programmer would

---

170. *Lotus*, 49 F.3d at 820 (Boudin, J., concurring).

171. Lotus Dev. Corp. v. Borland Int'l, 831 F. Supp. 223, 226 (D. Mass. 1993), *rev'd*,
49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).
Both 1-2-3 and Quattro Pro include sophisticated macro programming languages in-
tended to enhance the functionality provided by the user's menu of commands. *See,*
e.g., Lotus Development Corporation, User's Guide: Lotus 1-2-3 Release 4 Spread-
sheet for Windows 303-25 (1993) (describing the use and functionality of the 1-2-3
macro programming system).

172. *Lotus*, 49 F.3d at 809.

173. *Lotus*, 831 F. Supp. at 229, 240. Borland also asserted the defenses of waiver,
laches, and estoppel, which failed in the District Court and will not be considered
here. *Id.* at 235-40

174. *Id.* at 245.

175. *Lotus*, 49 F.3d at 819.

176. *See* Band, *supra* note 149, at 3 (noting the court's focus on 1-2-3's "command
structures").

177. *See* Baker v. Selden, 101 U.S. 99, 103 (1879); *supra* notes 34-41 and accompa-
nying text.

178. *See* Band, *supra* note 149, at 3.

be protected expression.[179] Under the approach taken by the district court, if an author has alternative ways of creating the expression, that expression is protectable.[180] Conversely, if only a limited number of practical ways of expressing a particular idea exist, the merger doctrine[181] prohibits copyright protection. For example, a menu that included only two commands, "Print" and "Exit," would not be protectable under the merger doctrine.

The 1-2-3 menu included 469 commands in over fifty menus and sub-menus creatively designed to represent complicated concepts in a way that the user would find accessible. The district court therefore concluded that the idea and expression had not merged and that copyright protected the expression.[182] Accordingly, the court enjoined Borland from selling any version of Quattro Pro that included either the menu emulation or the macro compatibility.[183]

In its decision, the district court rejected Borland's arguments that related to their defense of fair use.[184] Significantly, the court concluded that the fact that the 1-2-3 menu system may have become a de facto standard for spreadsheet menus would enhance rather than diminish the argument for copyright protection.[185] It reasoned that permitting protectable expression to lose its copyright protection by virtue of its success in the marketplace would be perverse.[186] Because commercially successful expression is the most likely to suffer from risks of infringement by competing products, the court concluded that it should view with considerable skepticism any argument for denial of protection on the grounds of commercial success.[187]

The district court considered only two players, Lotus and Borland, as relevant to its decision. It viewed any investment by the user of the software in the form of learning the menus and writing macros as falling within the scope of the copyright protection originally granted to the 1-2-3 program.[188] Thus, the district court concluded that the fact that the widespread use of a particular program may result in a de facto industry standard only implicates the success of the expressions

---

179. Lotus Dev. Corp. v. Borland Int'l, 831 F. Supp. 202, 215 (D. Mass. 1993), *rev'd*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

180. Band, *supra* note 149, at 3.

181. *See supra* notes 42-44 and accompanying text.

182. *Lotus*, 831 F. Supp. at 233.

183. *Id.* at 245.

184. *Id.*

185. *See* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 79 (D. Mass 1990). The *Paperback* case was a prior case involving copyright protection of the 1-2-3 program, which was decided by Judge Keeton of the District of Massachusetts, who also presided over the *Lotus v. Borland* trial. *Id.* at 37.

186. *Id.* at 79; *see also* Band, *supra* note 149, at 3 ("Copyright should reward creativity, . . . not penalize it.").

187. *See* Band, *supra* note 149, at 3.

188. *Id.* at 3-4.

contained in that product, and not any investment by the program users.

B.   *The Circuit Court Opinion*

On March 9, 1995, the First Circuit reversed the district court, holding that the 1-2-3 menu is not protectable under copyright law.[189] It held that the 1-2-3 menu structure was an unprotected "method of operation" under § 102(b)[190] of the Copyright Act.[191] It rejected the district court argument that the large variety of ways to construct a spreadsheet menu meant that the expression was protectable.[192]

Underlying the decision was an understanding of the interoperability issues raised by Borland's use of the 1-2-3 menu, particularly in the context of the macro compatibility. The circuit court distinguished between the areas over which copyright is supposed to grant a monopoly and the areas over which it is not. It recognized that prohibiting Borland's use of the 1-2-3 menu structure to allow Quattro Pro to play user-created macros would essentially expand the scope of the 1-2-3 copyright into an area of expression that belonged entirely to the user who created the macro.[193] The circuit court also recognized the importance of the user investment in learning the 1-2-3 menu and incorporating it into macros, stating that "[u]nder Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. . . . We find this absurd."[194] Accordingly, the circuit court denied all copyright protection to the 1-2-3 menu.[195]

V.   A MARKET-ORIENTED APPROACH TO *LOTUS V. BORLAND*

This part analyzes *Lotus* from an interoperability perspective, accounting for both the infringement concerns of the district court and interoperability concerns expressed by the circuit court. Part V.A argues that the circuit court was incorrect in holding that the 1-2-3 menu is an unprotectable method of operation. Part V.B evaluates the protectability of the 1-2-3 menu using the *Altai* test, concluding that the district court was correct in holding that the 1-2-3 menu is copyright-

---

189. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 819 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).
190. Section 102(b) of the Copyright Act provides:
> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation . . . regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b) (1994).
191. *Lotus*, 49 F.3d at 816.
192. *Id.*
193. *Id.* at 818.
194. *Id.* at 817-18.
195. *Id.* at 815.

able subject matter. Finally, part V.C applies the fair use doctrine in a manner that addresses the interoperability issues raised by the facts of Borland's use of the 1-2-3 menu.

A. *The 1-2-3 Menus Are Not Buttons*

In the process of reaching its reversal of the district court, the First Circuit drew an analogy between the 1-2-3 menu commands and the buttons on the front of a video cassette recorder ("VCR"). The court stated:

> Just as one could not operate a buttonless VCR, it would be impossible to operate Lotus 1-2-3 without employing its menu command hierarchy. Thus the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves. . . . Without the menu commands, there would be no way to "push" the Lotus buttons, as one could push unlabeled VCR buttons.[196]

According to the court, this analogy supports its argument that the 1-2-3 menu system is in fact an "uncopyrightable 'method of operation.' "[197] This analogy is incorrect. The protection sought involves the 1-2-3 labels, and *not* the buttons themselves.

Just as two different VCRs perform many identical functions, no dispute existed that 1-2-3 and Quattro Pro perform many identical functions. Lotus did not argue that the Quattro Pro menu buttons themselves cannot access functions identical to those of 1-2-3. If that were so, Lotus could have brought suit against any spreadsheet manufacturer on the ground that the "buttons" of all spreadsheet programs often duplicate 1-2-3 functions.[198] Lotus in fact argued that the but-

---

196. *Id.* at 817.

197. *Id.* at 815. In some sense, the court apparently relied on the plain meaning of the words "method of operation" to justify their holding. They stated that "[t]he Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3." *Id.* This plain-meaning interpretation of the words "method of operation" is incorrect. In the same sense, a computer operating system program is also a method of operation with regard to the machine that it operates, yet computer operating system programs are clearly protected by copyright. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1250-51 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984). For copyright purposes, it should not matter whether an expressive element of a computer program also happens to be a method of operating that computer program. This type of literal interpretation of terms of art should generally be viewed with some skepticism, particularly in the area of copyright law. *See, e.g.*, Lotus Development Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 56-58 (D. Mass. 1990) (declining to employ a literal interpretation of the "useful article" provision in the copyright statutes).

198. To prevent such a result, the copyright law includes an exception for "useful articles" or "utilitarian objects" that would preclude copyright protection for "menu commands," "mouse buttons," and the like. 17 U.S.C. § 101 (1994) (defining "useful article"); *see Paperback*, 740 F. Supp. at 54-55.

tons accessing identical functions in Quattro Pro may not feature the identical labels as the buttons in 1-2-3.[199]

Undoubtedly, the 1-2-3 menu designers sought to identify and organize the 469 functions of the spreadsheet in a way that communicates their purpose and use most effectively to the user. A successful menu system demands a significant amount of originality, effort, and expense to design.[200] Indeed, the ease with which a complicated computer program such as a spreadsheet can be learned and used carries considerable weight in the consumer's purchasing decision.[201] A term often used to describe this factor is "user friendliness."[202]

Despite Borland's stated goal of facilitating interoperability, the potential is present that the use of the 1-2-3 menu could result in a free ride for Quattro Pro users who find that the 1-2-3 menu is easier to work with than the Quattro Pro menu.[203] Consumers may view Quattro Pro as providing two programs for the price of one. When viewed in this light, Lotus's claim of infringement does not seem as disingenuous as the First Circuit suggested.

Under the circuit court's reasoning, nothing requires a competitor even to bother to offer its own menu in addition to including the Lotus menu. Under this "method of operation" rule, any program that has enjoyed success in the marketplace is open to duplication of its unique system of menus without the remedy of an infringement lawsuit. A hypothetical competitor could even legally market its product as "like Lotus—only cheaper," by virtue of the menu compatibility. This was the case in *Lotus Development Corp. v. Paperback Software International*,[204] where Paperback was marketing a spreadsheet program called VP-Planner, which was "publicly advertised as a 'workalike for 1-2-3.' "[205] In fact, the VP-Planner manual declared: "VP-Planner is designed to work like Lotus 1-2-3, keystroke for keys-

---

199. *Lotus*, 49 F.3d at 810. The circuit court opinion makes much of the fact that the Lotus menu command hierarchy is unique in that it is actually incorporated into the structure of the 1-2-3 program code. *See id.* at 815-16. This incorporation occurs in such a way that it is not possible to access the functionality of the 1-2-3 code without employing the menu command hierarchy in some form. *Id.* This discussion is irrelevant, however, because Borland is not charged with copying any Lotus code or with incorporating this particular programming technique in the Quattro Pro program. A peculiarity of program structure should not affect the method of copyright analysis or the result.

200. Samuelson, *Counterpoint, supra* note 84, at 13.

201. *See Manifesto, supra* note 7, at 2374-75; *see also* Clapes, *supra* note 2, at 91-92 ("[A] program's user interfaces determine the degree of the program's user friendliness, a highly qualitative characteristic that has a substantial impact on marketplace success.").

202. Clapes, *supra* note 2, at 92.

203. *Cf.* Samuelson, *Counterpoint, supra* note 84, at 13 (discussing the economic policy problems associated with "free riders," those that unfairly benefit when copyright provides inadequate protection of valuable components of computer software).

204. 740 F. Supp. 37 (D. Mass. 1990).

205. *Id.* at 69.

troke."[206] Unsurprisingly, the court found Paperback liable to Lotus for infringement.[207] It is hard to believe that the First Circuit intended to legitimize such conduct.

If computer program menus are truly unprotectable "methods of operation," the motivation of the software industry to continue to dedicate significant resources to the creation of "user friendly" menus will be seriously reduced, if not eliminated.[208] Such an interpretation of the copyright law "would stifle the very creativity that the law is designed to foster."[209]

B. *The 1-2-3 Menu: Protected Expression Under* Altai

If the 1-2-3 menu is actually about labels, and not just a set of buttons as the circuit court suggested, then it is not merely a "method of operation." At least five other federal circuits are at odds with the "method of operation" holding of the First Circuit in *Lotus*.[210] The Ninth Circuit has stated that "screens, *menus*, and *keystrokes*" are proper areas for copyright protection.[211] The Fifth Circuit held in *Engineering Dynamics, Inc. v. Structural Software, Inc.*[212] that the 230 "input formats"[213] of a structural engineering program "taken as a whole, readily qualify as [protectable] 'expression.' "[214] The court drew a parallel between the expressive characteristics of the 1-2-3 menu and the disputed input formats that it held to be protected expression.[215] The First Circuit in *Lotus* was apparently more concerned

---

206. *Id.* at 69.

207. *Id.* at 84.

208. *See* Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1344 (5th Cir. 1994) ("[I]f a best-selling program's interface were not copyrightable, competitors would be free to emulate the popular interface exactly . . . . This cannot be the law."); Manifesto, *supra* note 7, at 2365-66, 2379 (noting that by copying program behavior, a "second comer can produce a market substitute with relatively trivial effort," thus resulting in market destructive effects).

209. Stewart v. Abend, 495 U.S. 207, 236 (1990) (citing Iowa State Univ. Research Fund., Inc v. American Broadcasting Cos., 1621 F.2d 57, 60 (2d Cir. 1980)).

210. Petition for a Writ of Certiorari, Lotus Dev. Corp. v. Borland Int'l, Inc., 116 S. Ct. 39 (No. 94-2003), *reprinted in* Reback, *supra* note 167, at 213. "The First Circuit's relatively brief opinion comes as a jarring departure from the prevailing consensus . . . . The First Circuit quarrels with everybody: with the district court, with the Nimmer treatise, with Learned Hand, and with the Second, Ninth, and Tenth Circuits." Reback, *supra* note 167, at 212.

211. Brown Bag Software, Inc. v. Symantec Corp., 960 F.2d 1465, 1477 (9th Cir. 1992) (citing Shaw v. Lindheim, 919 F.2d 1353 (9th Cir. 1990) (emphasis added)).

212. 26 F.3d 1335 (5th Cir. 1994).

213. The "input formats" were essentially methods of entering complex arrangements of data into a computer program that was designed to solve structural engineering problems. *Id.* at 1338-39.

214. *Id.* at 1344.

215. *Id.* at 1345-46 (stating that the 1-2-3 menu "should be copyrightable because as a whole, the interface's structure and hierarchy constitute a high degree of original expression").

with achieving a result that permitted Borland's conduct than it was with following this significant body of caselaw.[216]

If the 1-2-3 menu is not a "method of operation," the question still remains whether it is copyrightable. As in most cases involving nonliteral copying[217] of computer programs, the initial inquiry attempts to separate the protectable elements from the unprotectable ones.[218] Under *Altai*, this analysis takes place through the first two steps of "abstraction" and "filtration."[219] Whatever expression is left protected is then "compared" to the challenged work to see if a "substantial similarity" exists between the two.[220]

1. Abstraction

In the abstraction step, a court retraces the steps followed by the original creator of the program, beginning with the highest level of abstraction, and proceeding to the lowest.[221] Generally, the highest level of abstraction would describe the general purpose that the program seeks to serve.[222] The lowest level of abstraction would be the actual program code that causes the program to function.[223]

In the case of the 1-2-3 program, the highest level of abstraction could be described as "an electronic spreadsheet." This level constitutes an unprotectable idea, not a protectable expression. A court would not seriously entertain a claim that the first copyrighted spreadsheet precluded the creation of subsequent spreadsheet programs.[224] In contrast, the actual 1-2-3 program code clearly falls within the realm of protectable expression, and not unprotectable idea. This principle has been confirmed by a number of cases that hold that the literal copying of program code constitutes copyright infringement.[225]

---

216. *See* Band, *supra* note 149, at 6.

217. Notably, the First Circuit in *Lotus* did not consider the case to involve nonliteral copying of program elements (the domain of *Altai*) despite the fact that none of the 1-2-3 code was copied. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 814-15 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996). The Court contended that the case was about the "literal copying of . . . a menu command hierarchy." *Id.* It is possible that this distinction was made to sidestep the *Altai-Whelan* debate and avoid Supreme Court review. *See* Band, *supra* note 149, at 7.

218. *See* Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 706 (2d Cir. 1992).

219. *Id.* at 706-10; *see supra* notes 96-101 and accompanying text.

220. *Id.* at 710.

221. *Id.* at 697-98, 706-07.

222. *See id.* at 707.

223. *Id.*

224. Lotus was not the first company to employ the idea of a computerized electronic spreadsheet. VisiCalc, a program that dominated the PC market in its early stages, was the first program to capitalize on the utility of this concept. *See* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp 37, 65 (D. Mass. 1990).

225. *See, e.g.*, Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253-54 (3d Cir. 1983) (stating that computer program in object code format should be protected by copyright), *cert. dismissed*, 464 U.S. 1033 (1984); Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 872-73 (3d Cir. 1982) (holding that computer program instruction codes are protected by copyright).

The 1-2-3 menu at issue in the *Lotus* case belongs in neither the highest nor the lowest category of abstraction.[226] The question therefore becomes, at what point along the abstraction spectrum do the specific labels and structure of the 1-2-3 menu system fall? Are they more like the "idea" of a computerized spreadsheet, or more like the program code that expresses the idea?

If the program concept is the idea, and the program code is an expression of that idea, then a court can draw a parallel to the "idea" of using a system of user-selected menu options to operate a computer program, and the resulting menu as expressed in a particular program. Just as there are many ways to express the idea of a computer spreadsheet program, there are many ways to express the idea of a spreadsheet program menu. The idea of a spreadsheet menu is a concept that is aimed at creating an intuitive and accessible connection between the user and the machine. The 1-2-3 menu itself is an original expression of that idea in a specific instance.[227]

The protectable nature of the 1-2-3 menu becomes more apparent when the menu is viewed as a whole, and not just the sum of its parts.[228] Specifically, "If particular characteristics not distinctive individually have been brought together in a way that makes the 'whole' a distinctive expression of an idea . . . then the 'whole' may be copyrightable."[229] Thus, even assuming that none of the individual menu commands was copyrightable, the "selection and arrangement" of the 469 commands in the 1-2-3 menu is eligible for copyright protection, because it is both creative and original.[230] At least in principle, then, the 1-2-3 menu system may be protected by copyright because it is more "expression" than "idea."

2. Filtration

Granting the protectable nature of a particular element of a computer program under the abstraction step, the next task is to filter out any parts of that element that are outside the scope of copyright pro-

---

226. Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1344 (5th Cir. 1994) ("In the middle of the abstraction spectrum sit user interfaces such as that of Lotus 1-2-3 . . . .").

227. This reasoning is consistent with the case law in a number of circuits. *See, e.g.,* *id.* at 1343-44 (comparing the 1-2-3 menu system with a system of 230 input-output formats to hold the latter system to be protectable expression and not unprotectable idea); Autoskill Inc. v. National Educ. Support Sys., Inc., 994 F.2d 1476, 1492 (10th Cir. 1993) (upholding the district court's determination that originality of organization, structure, and sequence results in protectable expression under the abstraction analysis); Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173, 1175-76 (9th Cir. 1989) (same).

228. *Paperback,* 740 F. Supp. at 67.

229. *Id.*

230. *See* Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 348-49 (1991).

tection.[231] The court must go beyond the idea-expression dichotomy to determine whether the 1-2-3 menu system warrants protection.

The *Altai* court focused on three categories of elements that must be denied protection in the filtration process: (1) elements dictated by efficiency; (2) elements dictated by external factors; and (3) elements taken from the public domain.[232] These three categories within the filtration process fall under the doctrines of merger and *scènes-à-faire*.[233]

As observed above, the merger doctrine would deny protection to a menu that consisted only of the items "Print" and "Exit."[234] After all, there are only so many ways of expressing those concepts. Although "Print" and "Exit" are both included in the 1-2-3 menu and are filtered out, these items are only two of 469 selections that the user may make.[235] The district court correctly noted that the stunning variety of representations of 469 menu selections arranged in fifty different menus precludes a finding that the 1-2-3 menu system was entirely the product of functional constraints.

Similarly, the *scènes-à-faire* doctrine will prohibit protection for "incidents, characters or settings which are as a practical matter indispensable, or at least standard, in the treatment of a given topic."[236] The *scènes-à-faire* doctrine applied to the 1-2-3 menu would certainly deny protection to menu items and characteristics that are common to most computer program menus. For example, like most modern menu-driven programs, the 1-2-3 for Windows menu includes as the first three menu headings the words File, Edit, and View.[237] This arrangement is probably unprotected under *scènes-à-faire* doctrine even though not dictated by functional constraints.

The filtration stage, therefore, removes from protection the components of the 1-2-3 menu that are dictated by function, and the components that are dictated by convention. Only a small fraction of the overall 1-2-3 menu system is filtered out in this stage. The majority of the menu remains as protected expression. Indeed, the remaining portion is probably what motivated Lotus to file the lawsuit.

---

231. *See* Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 707-10 (2d Cir. 1992).

232. *Id.*

233. *See Engineering Dynamics*, 26 F.3d at 1344-45; *see also supra* notes 42-47, and accompanying text (discussing merger and *scènes-à-faire* doctrines).

234. *See supra* notes 181-82 and accompanying text.

235. The Circuit Court cited the commands "print" and "copy" as proof of the unprotected nature of the 1-2-3 menu, but never addressed the protectability of the 1-2-3 menu as a whole. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 815 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

236. Atari, Inc. v. North Am. Philips Consumer Elec. Corp., 672 F.2d 607, 616 (7th Cir. 1982) (citing Alexander v. Haley, 460 F. Supp. 40, 45 (S.D.N.Y. 1978)).

237. Lotus Development Corporation, User's Guide: Lotus 1-2-3 Release 4 Spreadsheet for Windows 10 (1993).

3. Comparison

The court determining the infringement issue must now compare the remaining elements of the 1-2-3 menu with the corresponding elements in the Quattro Pro program. According to *Altai*, two determinations take place at this stage. First, the court must determine if "substantial similarity" exists between the remaining protected expression and the allegedly infringing program.[238] Second, the court must make "an assessment of the copied portion's relative importance with respect to the plaintiff's overall program."[239]

A strong argument can be made that the level of similarity required for infringement should depend in part on the range of expression available to the copyrighted and allegedly infringing works.[240] A work of fiction, for example, should not require as much similarity for a finding of infringement as would a purely factual work.[241] Because all spreadsheet menus are similar to the extent that they are expressing a similar idea, a high standard of similarity should be required for a finding of infringement.[242] Because Borland concedes that its use of the 1-2-3 menu is a verbatim use, the conclusion is inescapable that the menus are substantially similar.[243]

As for the second determination, a finding of substantial similarity will be dispositive unless the amount copied is so small as to be insignificant.[244] In this case, not only was the 1-2-3 menu copied in its entirety, but the menu is qualitatively important to the 1-2-3 program.[245] Thus, under the *Altai* test, Borland's use of the 1-2-3 menu infringes the Lotus copyright. Borland should therefore be liable unless it can invoke the affirmative defense of fair use.

---

238. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 710 (2d Cir. 1992).

239. *Id.*

240. *See, e.g.*, Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 349 (1991) (confirming the protectability of factual compilations, but stating that such protection is "thin").

241. *See id.* at 348-350.

242. In contrast, a menu system designed to control a fictional video game would probably not require as much similarity to justify a finding of infringement, because the range of possible expression is greater. *See, e.g.*, Atari, Inc. v. North Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 619 (7th Cir. 1982) (considering the nature and setting of the works and finding substantial similarity between two fictional video games notwithstanding the fact that significant differences existed between the two).

243. *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp 223, 234 (D. Mass. 1993) ("[W]hat was copied by Borland . . . is virtually identical to [the] expression of the Lotus 1-2-3 program's menu structure."), *rev'd*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996). Even under the "virtual identity" standard followed in Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp 616, 625 (N.D. Cal. 1993), Borland's use would constitute infringement.

244. 3 Nimmer on Copyright, *supra* note 24, § 13.03[F], at 13-148.

245. *See* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp 37, 68 (D. Mass. 1990).

C.   *The Fair Use Doctrine Can Properly Evaluate Interoperability Issues*

If the 1-2-3 menu enjoys copyright protection and Borland has infringed that copyright, then the best available means of evaluating Borland's interoperability arguments is through the fair use doctrine.[246] The doctrine is most naturally applicable where the defendant concedes the use of plaintiff's material, but argues that the court should not construe the law to prohibit such a use.[247] As noted above, the fair use doctrine is codified by statute and provides for the balancing of four factors to determine if a particular use of protected material is fair.[248] This part will now evaluate the four factors with regard to the facts of Borland's use of the 1-2-3 menu, focusing particularly on interoperability issues.

1.   The Purpose and Character of the Use

Until recently, much of the caselaw that has interpreted this factor has focused on whether the disputed use of the copyrighted material is for commercial purposes.[249] In this case, Borland's use of the 1-2-3 menu is unquestionably commercial, a fact that weighs against a finding of fair use.[250] The district court rejected Borland's fair use defense, stating that "when a defendant's use of copied materials is determined to be commercial, that use is 'presumptively an unfair exploitation of . . . the copyright.' "[251]

This reasoning does not reflect a proper interpretation of the statute.[252] The Supreme Court has recently stated that "[t]he language of the statute makes clear that the commercial . . . purpose of a work is only one element of the first factor enquiry into [the use's] purpose and character."[253] Thus, to evaluate this factor properly, a court must consider the use from the perspective of the interoperability requirements of computer programs. To this effect, in a concurrence to the First Circuit opinion in *Lotus*, Judge Boudin criticized the district court's rejection of the fair use defense, stating: " '[P]resumptively' does not mean 'always' and, in any event, the doctrine of fair use was

---

246. *See, e.g.*, Raskind, *supra* note 51, at 1182 (recommending the application of the fair use doctrine to "serve as an arbiter between permissible use[s] of protected software . . . and impermissible piracy").

247. In the *Lotus* case, Borland argued both that the 1-2-3 menu was unprotected, and alternatively that their use of it was fair. *Lotus*, 831 F. Supp. at 229, 240-45.

248. 17 U.S.C. § 107 (1994); *see supra* notes 48-60 and accompanying text.

249. *See, e.g.*, Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 562 (1985) (distinguishing commercial uses from nonprofit educational uses); Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 451 (1984) (stating that commercial uses of copyrighted material are presumptively unfair).

250. *See* Patry, *supra* note 52, at 426.

251. *Lotus*, 831 F. Supp. at 242 (quoting Stewart v. Abend, 495 U.S. 207, 237 (1990) (citation omitted)).

252. *See* Patry, *supra* note 52, at 420-21.

253. Campbell v. Acuff-Rose Music, Inc., 114 S. Ct. 1164, 1174 (1994).

created by the courts and can be adapted to new purposes."[254] Judge Boudin suggested that the utilitarian nature of computer programs requires the modification of traditional copyright analysis to address new issues.[255]

a. *Interoperability As a Fair Use*

In addition to being utilitarian in nature, computer programs are also unique in copyright law in that they must be able to coexist in an organized, interoperable framework that allows communication between the user and the program, as well as between one program and another.[256] Interoperability requirements demand that the scope of the "temporary monopoly" granted by copyright protection be tailored to both protect the expression of the author and to not unreasonably restrict the investment, creativity, or expressions of others. In other words, a copyright monopoly granted in one "market"[257] should not exercise its power over other "markets."[258]

An example will illustrate how these software "markets" interact. Microsoft Windows is a popular operating system[259] that serves the personal computer operating systems market.[260] If the copyright protection for Microsoft Windows could prevent application programs,[261] like 1-2-3 or WordPerfect, from being viable products, then the Windows copyright has acted outside of the legitimate scope of protection that copyright should afford it.[262] This overprotection would occur if the specific data structures that allow the necessary communication between the Windows operating system and the application programs that run under Windows were held to be copyrightable.[263] The idea-

---

254. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 821 (1st Cir. 1995) (Boudin, J., concurring), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

255. *See id.* at 819-20 (Boudin, J., concurring).

256. *See Manifesto, supra* note 7, at 2375-76.

257. The term "market" is used here as an abstraction to describe the extent to which a particular copyrighted work meets a particular need.

258. *See, e.g.*, Band, *supra* note 149, at 6 (observing that a "monopolist has little incentive to innovate—the ultimate goal of the intellectual property system"); Gordon, *supra* note 143, at 868-69 (arguing for the desirability of competitive intellectual property markets and protection thereof); Warren-Boulton, *supra* note 150, at 18 (arguing that the leveraging of a copyright monopoly into other markets makes the rest of society worse off).

259. An "operating system" like Windows provides a software environment that allows the application programs to run. *See* Clapes, *supra* note 2, at 28.

260. *See Manifesto, supra* note 7, at 2377.

261. "Application programs" include word processors, spreadsheets, and database programs that interact directly with the user. They are to be contrasted with operating systems that allow the application programs to run. *See* Bernard A. Galler, Software and Intellectual Property Protection 50-51 (1995).

262. *See, e.g.*, Warren-Boulton, *supra* note 150, at 18 (arguing that allowing a copyright monopoly should not be used for anticompetitive purposes).

263. *See, e.g.*, Clapes, *supra* note 2, at 179-82 (noting that conformity of "internal formats and protocols" is necessary for compatibility).

expression distinction is one method of preventing such copyright abuses.[264]

The facts of *Lotus* are analogous. If Lotus can block macro compatibility by claiming copyright infringement, then the user-created macros are subject to the copyright monopoly held by Lotus, despite the fact that Lotus has contributed nothing to the expression incorporated into the user's macro.[265] Similarly, Lotus would hold a monopoly over the user's investment in learning the menus of 1-2-3, an investment which is not a result of any contribution by the Lotus programmers.

Judge Boudin's concurring opinion noted this fact: "If Lotus is granted a monopoly on this [menu] pattern, users who have learned the command structure of Lotus 1-2-3 or devised their own macros are locked into Lotus . . . ."[266] Under this reasoning, uses of copyrighted material that facilitate interoperability should be more likely to be fair uses because they prevent the unwarranted extension of the copyright monopoly into separate markets.

b. *Transformative Works*

Borland's use of the menu is also supported by the fact that the Quattro Pro program is transformative.[267] To determine whether a work is transformative, the court asks if the work merely "supplant[s] the original . . . or instead adds something new, with a further purpose or different character."[268] The circuit court noted that "Borland's objective was to develop a spreadsheet program far superior to existing programs, including Lotus 1-2-3."[269]

Quattro Pro does not merely duplicate the functionality of 1-2-3—it supersedes it: "[T]he goal of copyright, to promote science and the arts, is generally furthered by the creation of transformative works."[270] If Borland had marketed Quattro Pro solely as a cheap version of 1-2-3, its use of the 1-2-3 menu would not be transformative; it would be more like theft.

Thus, the first fair use factor should weigh in favor of Borland. Despite Quattro Pro's commercial nature, the character and purpose of its use of the 1-2-3 menu legitimately serves to facilitate the interoperability of user-created macros and investment in learning the menu.

---

264. Warren-Boulton, *supra* note 150, at 18.

265. *See, e.g.*, Lewis Galoob Toys v. Nintendo of Am., 780 F. Supp. 1283, 1297 (N.D. Cal. 1991) (arguing that the commercial success of a copyrighted product does not warrant the extension of protection beyond that granted by the law).

266. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 821 (1st Cir. 1995) (Boudin, J., concurring), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

267. *See* Band, *supra* note 149, at 5-6.

268. Campbell v. Acuff-Rose Music, Inc., 114 S. Ct. 1164, 1171 (1994).

269. *Lotus*, 49 F.3d at 810.

270. *See Acuff-Rose Music*, 114 S. Ct. at 1171.

Moreover, its functionality goes beyond that of 1-2-3, making it a "transformative work."

2. The Nature of the Copyrighted Work

The second fair use factor recognizes that copyrighted works receive differing levels of protection depending on their type.[271] This differentiation occurs because "the law generally recognizes a greater need to disseminate factual works than works of fiction or fantasy."[272] Thus, works with strong factual characteristics, such as the 1-2-3 menu, are copyrightable, but the copyright in such a work is "thin."[273]

The district court argued that even for functional works, where the author enjoys a great deal of freedom in creating his expression, the scope of the protection afforded should be greater.[274] In the case of the 1-2-3 menu, however, this argument is unpersuasive because despite the creative features of the 1-2-3 menu, the degree of creativity is constrained by the function that the menu is designed to serve. Similar restrictions are faced by all authors of spreadsheet menus, and the commands the menus access are frequently identical. Thus, in the spectrum of fictional works to factual works, the 1-2-3 menu is more fact than fiction.[275] As a result, this factor also weighs in Borland's favor.[276]

3. The Amount and Substantiality of the Portion Used

The third fair use factor evaluates the significance of the copied material to the copyrighted work. The evaluation not only looks at what percentage of the copyrighted work is used, but also looks to the relative importance of the copied portions.[277] Even where only small portions of the copyrighted work are used, if they constitute the "heart" of the work, the factor must weigh against finding a fair use.[278]

---

271. Patry, *supra* note 52, at 504-07.

272. Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 563 (1985).

273. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1524 (9th Cir. 1992) (citing Feist Publications, Inc. v. Rural Tel Serv. Co., 499 U.S. 340, 349 (1991)).

274. Lotus Dev. Corp. v. Borland Int'l, 831 F. Supp. 223, 242-43 (D. Mass. 1993), *rev'd*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

275. Computer programs are usually classified as "factual works." Patry, *supra* note 52, at 505-06 & n.483.

276. Some commentators caution against a "mechanical weighing of the second factor against the [computer software] copyright owner." *Id.* at 519-20. Considering the similar nature of all spreadsheet programs and their menu systems, however, it is appropriate to conclude that the second factor was established to draw a distinction between restricted expression such as the 1-2-3 menu and the unrestricted expression typical of works of fiction.

277. *See id.* at 549-51.

278. Harper & Row, Publishers, Inc. v. Nation Enters., 557 F. Supp 1067, 1072 (S.D.N.Y. 1983) (Owen, J.) (citation omitted).

As observed above, the design of the menu system is a significant feature of a complicated computer spreadsheet program.[279] It is one of the key elements in determining how easy a program is to use, and how attractive the program will be to the consumer.[280] Considering the importance of the menu to the 1-2-3 program and the fact that Borland copied the menu entirely, this factor must weigh in favor of Lotus.

4. The Effect of the Use Upon the Potential Market for the Copyrighted Work

The fourth factor evaluates the economic impact of the challenged use on the copyrighted work.[281] In many respects, this factor is the most significant aspect of the fair use evaluation.[282] It is not, however, merely a superficial analysis that determines the loss in value of the copyright resulting from the use. In at least some sense, every legally permissible use of copyrighted material decreases the value of the copyright.[283]

In analyzing the fourth fair use factor, the court must draw distinctions between "remediable" and "unremediable" harm.[284] By making it easier for 1-2-3 users to switch to Quattro Pro, Borland's use would decrease the value of the 1-2-3 copyright. If the protection Lotus seeks goes beyond the proper scope of its copyright, however, the resulting decrease is not of a type that should weigh against a finding of fair use.[285] Thus, in order to evaluate the market effects properly, a court should consider the impact of Borland's use on the purchasing decisions made by consumers.

Although the retail cost of today's packaged software has fallen considerably when compared with the software of the past, "measuring the true cost of acquiring software requires taking into account the cost of *using* a new product."[286] For example, a typical employee earning $40,000 a year costs a company fifty dollars per hour at

---

279. *See supra* notes 200-02 and accompanying text.

280. *See* Clapes, *supra* note 2, at 91-92.

281. *See* Patry, *supra* note 52, at 561-63.

282. *Harper & Row*, 471 U.S. at 566; *see also* Campbell v. Acuff-Rose Music, Inc., 114 S. Ct. 1164, 1177 (1994) (holding that a fair use proponent "would have difficulty carrying the burden of demonstrating fair use without favorable evidence about relevant markets").

283. *See* Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 820 (1st Cir. 1995) (Boudin, J., concurring) (noting that "any use of the Lotus menu by Borland is a commercial use and deprives Lotus a portion of it's 'reward," but stating that "this is circular reasoning"), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

284. *See Acuff-Rose Music*, 114 S. Ct. at 1178 (stating that copyright law distinguishes between "remediable" and "unremediable" harm to the market for the original work).

285. *See, e.g., id.* at 1178-79 (holding that damage to an unprotectable market does not weigh against a finding of fair use).

286. *Manifesto, supra* note 7, at 2374 (emphasis added).

work.[287] Thus, the time spent by the employee learning to use a new software package can easily cost more to the company than the purchase of the software does.[288] Given these realities, the significant economic barriers inherent in switching from one type of spreadsheet to another (once the investment in learning and developing macros has been made) are clear.

But what about the company that is looking to make its first investment in an electronic spreadsheet? For them, no prior investment in learning requires protection. The primary considerations of the initial purchasing decision will be (1) the retail cost of the software and its functionality, and (2) the cost and ease of learning to use the software and integrating it into the company's business.

This difference in purchasing considerations highlights a key fact of Borland's defense of interoperability. The decision about whether to switch from 1-2-3 to Quattro Pro is different from the decision to make an initial purchase of Quattro Pro as opposed to 1-2-3. Borland's interoperability defense only applies to the former decision and not to the latter. This is because no existing user investments prevent the first-time buyer from choosing the electronic spreadsheet that is most suitable.[289]

The perspective of the first-time buyer also points out a key difference between Borland's use of the 1-2-3 menu for (1) the purpose of macro compatibility, and (2) as part of the Quattro Pro user interface. The macro compatibility feature is of no consequence to the first-time buyers of an electronic spreadsheet, because they have no macros yet.

By contrast, a company that is going to invest in an electronic spreadsheet for the first time may in fact give significant weight to the fact that the 1-2-3 menu system is included with Quattro Pro in addition to the Quattro Pro menu.[290] This feature will be important for at least two reasons. First, the buyers of Quattro Pro in essence get two programs for the price of one because the users can then choose between the two menu systems to find the one that works best for them. Second, any new employees who join the company at a later time who are familiar with either Quattro Pro or 1-2-3 will be accommodated by the purchase of Quattro Pro, but not by the purchase of only 1-2-3. This analysis leads to three conclusions with regard to interpreting the fourth fair use factor.

---

287. *Id.*

288. *Id.* at 2374-75; *see also* Farrell, *supra* note 148, at 37 n.3 ("[O]ne consultant estimates that training costs for a popular spreadsheet program are approximately one thousand dollars per user.").

289. These "market segment" differences will arise in any circumstance where a competitor seeks to incorporate components of a "de facto" industry standard to facilitate interoperability.

290. *See supra* notes 200-02 and accompanying text.

First, losses to Lotus that arise out of facilitating a switch to Quattro Pro by existing 1-2-3 users do not weigh against a finding of fair use. Those considering a switch from 1-2-3 to Quattro Pro face the "market barriers"[291] associated with having invested time and effort to use 1-2-3 and write macros for it. The users of 1-2-3 are reluctant to switch to Quattro Pro because they will have to throw away their investment in learning the 1-2-3 menu system. Granting such protection to Lotus would coopt the investment by the public in its program. Copyright protection should not exercise its monopoly power over this investment by the user. Therefore, removing these market barriers does not result in any remediable harm to Lotus.

Second, the use of the 1-2-3 menu to achieve macro compatibility in Quattro Pro does not result in any additional decrease to the value of 1-2-3 copyright. Because a first-time buyer of an electronic spreadsheet program has not written any macros for 1-2-3, no additional injury to Lotus takes place when first-time buyers are added into the market analysis. Therefore, the only effect of this particular use is the removal of the market barriers faced by owners of 1-2-3 who have invested time and effort writing macros for 1-2-3.

Third, the use of the 1-2-3 menu as part of the Quattro Pro user interface allows an unwarranted free ride on the copyrighted expression of 1-2-3 for first-time buyers of electronic spreadsheets. Consumers who have not invested time and effort learning to use 1-2-3 may find Quattro Pro to be a more attractive program because of the inclusion of the 1-2-3 menu as an option in Quattro Pro. To this extent, the increase in the attractiveness of Quattro Pro comes entirely at the expense of the 1-2-3 copyright.

Given these conclusions, the fourth fair use factor must weigh in Borland's favor with regard to the macro compatibility use, and in favor of Lotus with regard to the use of the 1-2-3 menu in the Quattro Pro user interface.

D. *Evaluation of the Fair Use Factors*

To protect computer software from cheap "knockoffs," a complicated system of menus must have some copyright protection. To call the 1-2-3 menu a "method of operation" is an oversimplification. Great creative effort is expended to design program menus in an accessible way.[292] The result of this creativity is precisely the kind of expression that copyright law is supposed to protect.[293]

---

291. The term "market barrier" is used here to describe the artificial reluctance of existing 1-2-3 users to switch to Quattro Pro, when they would otherwise seek to employ some additional functionality provided by Quattro Pro. *Cf. Manifesto, supra* note 7, at 2412 (discussing artificial barriers to market entry by manufacturers).

292. Samuelson, *Counterpoint, supra* note 84, at 13.

293. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 67 (D. Mass. 1990).

The balance of the four fair use factors strongly supports a finding that the use of the 1-2-3 menu to facilitate macro compatibility is a fair use under § 107. The first, second, and fourth factors all support this use. It does not have an unfair negative impact on the market for 1-2-3 because only those consumers who have written their own macros will be affected by the use. Borland should therefore be permitted to incorporate the 1-2-3 menu in the Quattro Pro program for the purpose of providing macro compatibility.

The use of the 1-2-3 menu in the Quattro Pro user interface presents a more difficult problem. Only the first and second factors support the use, with the third and fourth weighing against it. This result requires a balancing of the positive effects resulting from interoperability with the negative effects arising from the loss of legitimate copyright protection to determine whether the use is fair.[294] Arguments exist on both sides. Nonetheless, it seems that where the unwarranted harm to the copyrighted work arising from the use is substantial, as it could be here, even a strong argument for fair use must fail if copyright protection is to be meaningful.[295]

VI. APPLICATION OF MARKET ANALYSIS TO *LOTUS*: A LIMITED FAIR USE

The basic problem with Borland's use of the 1-2-3 menu in the Quattro Pro user interface is that it gives Borland access to a free ride from first-time spreadsheet purchasers who choose Quattro Pro. Borland benefits unfairly at the expense of Lotus when consumers who do not already own 1-2-3 purchase Quattro Pro because of the inclusion of the 1-2-3 menu. If a court can calibrate protection to minimize the unwarranted harm to Lotus, perhaps Borland's use of the 1-2-3 menu as part of the Quattro Pro user interface could qualify as a fair use.[296]

Because the harm to the market for 1-2-3 only manifests itself in the purchasing decision of consumers who do not already own a copy of 1-2-3, any resolution should focus on denying Borland access to that market.[297] One means of accomplishing this would be to require Borland to establish proof of ownership of a copy of 1-2-3 by the consumer before the 1-2-3 menu emulation could be enabled in Quattro Pro. Such proof could consist of a copy of the 1-2-3 license included with the program. Once a Quattro Pro consumer established owner-

---

294. *See* Raskind, *supra* note 51, at 1182.

295. *See* Rogers v. Koons, 960 F.2d 301, 312 (2d Cir. 1992) (noting that "unfair copying undercuts demand for the original work and, as an inevitable consequence, chills creation of such works").

296. *See, e.g.*, Raskind, *supra* note 51, at 1181-82 (arguing for the extension of fair use to reflect industry practices, maintain the incentive to create, and avoid diminishing the competitive value of copyrighted works).

297. The relevant market would be the first time buyers of electronic spreadsheets that desired access to the 1-2-3 menu.

ship of the 1-2-3 program, Borland could enable the 1-2-3 menu emulation either by providing the user with a password, or by shipping a computer disk that added the feature onto an installed copy of Quattro Pro.

Existing caselaw provides some support for this reasoning. In *Lewis Galoob Toys v. Nintendo of America*,[298] the Ninth Circuit addressed the question of whether a computer game cartridge that altered the performance of another copyrighted computer game resulted in an unauthorized "derivative work."[299] The court found that although the copyrighted program was being used in a manner unanticipated by its creators, the challenged use was fair under § 107.[300] Central to this holding was the conclusion that the primary market for the copyrighted work was not undermined by the creation of the derivative work.[301] Nintendo's market remained unaffected because Galoob's product was useless without the purchase of Nintendo's games.[302]

This Note suggests that the availability of the fair use defense to sanction the copying of user interface elements necessary for interoperability should be predicated on proof that the use was strictly limited to facilitating interoperability. Where the use has effects that go beyond facilitating interoperability to undermine the demand for the original unfairly, the defense should not be available. This approach will be called "limited fair use." The effects of such an approach will now be analyzed from the perspective of Lotus, Borland, and the public in general.

A. *Effect on Lotus*

Clearly Lotus would be better off under the limited fair use approach than it is under the "method of operation" rule of the First Circuit.[303] Under the "method of operation" rule, Lotus is not only vulnerable to uses of its menus by major competitors like Borland, but it is also vulnerable to "low-rent" competitors who manufacture "clones," whose only attraction is offering a cheaper version of 1-2-3.[304] The circuit opinion apparently endorses "keystroke for keys-

---

298. 964 F.2d 965 (9th Cir. 1992), *cert. denied*, 113 S. Ct. 1582 (1993).

299. *Id.* at 967.

300. *Id.* at 972. The plaintiff, Nintendo, had asserted that the use of defendant's product resulted in the creation of an unauthorized derivative work. *Id.* at 967. Although the Ninth Circuit dismissed the derivative work theory, it held that even if Galoob's product did create derivative works, Galoob would not be liable under the fair use doctrine. *Id.* at 972.

301. *Id.* at 971.

302. Lewis Galoob Toys v. Nintendo of Am., 780 F. Supp. 1283, 1294 (N.D. Cal. 1991).

303. *See supra* notes 189-95 and accompanying text.

304. Manifesto, *supra* note 7, at 2366 ("Firms that can rapidly replicate the innovative behavior of another firm's product at much lower development costs can supply a market substitute that will deny the innovator an adequate opportunity to recoup its

troke" compatibility claims of products like Paperback's VP-Planner.[305]

The limited fair use approach would effectively prohibit uses of the 1-2-3 menu system in programs like VP-Planner that are not trans-formative. This result arises because limited fair use is predicated on preexisting ownership; therefore, the clone market is restricted to those who already own the original, a very small market indeed.[306] Thus, under limited fair use, only transformative works will be viable candidates for incorporating the copyrighted menus.

In some sense, Lotus is in fact better off under limited fair use than it would have been had the district court opinion prevailed, granting full protection to the 1-2-3 menu and denying Borland's fair use de-fense.[307] Allowing transformative works to borrow elements neces-sary for interoperability creates another market for the original expression.[308] If access to that market is restricted to the original copyright holder, some increased demand for the original expression should result. The proceeds of the increased demand would flow di-rectly to Lotus.

Imagine a hypothetical company employing 100 people who all use 1-2-3. Imagine further that the company wants to switch to Quattro Pro to obtain additional functionality. Under the limited fair use ap-proach, if the company switched, it could obtain access to the 1-2-3 menu by virtue of the 100 1-2-3 licenses that it already owned. Now suppose that the company expanded its operation and added an addi-tional twenty people. In order to continue to enjoy both the 1-2-3 menu and the Quattro Pro functionality, it must purchase both prod-ucts. In this way, allowing consumer access to the 1-2-3 menu in Quat-tro Pro both expands the market for 1-2-3 and achieves interoperability at the same time.

B. *Effect on Borland*

Borland contended from the beginning that it included the 1-2-3 menu in Quattro Pro so that "spreadsheet users who were already familiar with Lotus 1-2-3 would be able to switch to the Borland pro-grams without having to learn new commands."[309] To contend other-wise would have resulted in an indefensible position. As noted in the

research and development expenses and make sufficient profits to justify its initial investment in innovation.").

305. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 69-70 (D. Mass 1990).

306. *See* Clapes, *supra* note 2, at 204-05.

307. Lotus Dev. Corp. v. Borland Int'l, 831 F. Supp. 223, 245 (D. Mass. 1993), *rev'd*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996); *see also supra* notes 177-88 and accompanying text (discussing the district court opinion).

308. *See Manifesto, supra* note 7, at 2371-72.

309. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 810 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

preceding section, Borland's stated purpose only applies to those who already own a copy of 1-2-3.[310] Thus, the limited fair use approach is consistent with the stated objective sought by Borland because it allows owners of 1-2-3 to switch without losing any personal investment in learning.[311]

In a strict sense, Borland achieves its goal under the limited fair use approach, with the exception that it faces some additional transaction costs associated with establishing customer ownership of a 1-2-3 license. Additional transaction costs would probably not be minimal considering the efforts of software companies to "register"[312] the owners of their software.[313] Moreover, many software companies already offer special "upgrade" discounts called "trade up editions" that are available to existing owners of competing products.[314] Because the availability of the fair use defense would depend on proper registration, the motivation to administer the system effectively would be high. In any event, any inconvenience is a small price to pay for the use of copyrighted material that allows the company to compete more effectively.

C. *Effect on the Public*

The primary concern expressed by the First Circuit in *Lotus* was that "if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. . . . We find this absurd."[315] The court concluded that once the users of 1-2-3 had become familiar with the 1-2-3 menu, to allow copyright law to prevent them from employing that knowledge to use other programs would be bad policy.[316]

---

310. *See supra* notes 286-89 and accompanying text.

311. Arguably, the owners of software licenses use the software, a fact which the limited fair use approach does not account for. In the end, however, only the owner of the software license determines the specifics of who can use the copy, as well as where and when it can be used.

312. Software registration is typically accomplished by the completion and return of a registration card included with the software. Users benefit from the registration of software because it typically entitles the user to new product information and may provide for software upgrades. *See* Thomas J. Smedinghoff, *Avoiding Liability for Copyright Infringement: Preventive Measures for Software Users*, 5 No. 11 J. Proprietary Rts. 2, 9 (1993).

313. Whether or not the transaction costs associated with the limited fair use approach would result in the underutilization of interoperability potential is debatable. When compared to the all or nothing approach taken by the district and circuit courts in *Lotus*, however, it seems that limited fair use is superior to both approaches in that it achieves interoperability and encourages creative development.

314. These upgrade discounts include with the software a program that electronically inspects the host computer to see if the competing software is actually resident in the computer before the new software is installed.

315. Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 817-18 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

316. *See Lotus*, 49 F.3d at 818.

The limited fair use approach, which requires Borland to establish the legitimacy of these consumer desires, does not undermine the First Circuit's concern for allowing interoperability. The only consumers denied access to the 1-2-3 menu are those who receive no interoperability benefits. Such consumers are therefore beyond the scope of the circuit court's concern.[317]

The district court, on the other hand, expressed a different concern, stating that de facto industry standards such as the 1-2-3 menu should not suffer the erosion of their copyright protection because of their success in the marketplace.[318] This policy also is not undercut by the approach described above. Indeed, the market for the 1-2-3 menu is entirely maintained because in order for the 1-2-3 menu to be used in Quattro Pro, the consumer must have already purchased a copy of 1-2-3.

The only distinction between the limited fair use approach and complete copyright protection would be that once the 1-2-3 menu is "owned"[319] by the consumer, the consumer may use it in a way that was not intended by the manufacturer. The fact that the use was unintended does not necessarily give the copyright holder any grounds for complaint.[320] Indeed, the fair use doctrine itself expressly sanctions unintended uses of copyrighted material.[321]

The public receives significant benefit from allowing unintended uses of copyrighted material that facilitate interoperability.[322] The benefits of interoperability fade quickly, however, when they are used to justify an attack on the primary market for copyrighted works. If interoperability becomes a justification for wholesale copying of important expressive elements of a successful program's user interface, then the motivation to innovate in this area will disintegrate.[323]

---

317. The exclusion of such consumers may actually prove to be beneficial. It has been suggested that a cost of interoperability lies in the loss in efficiency that arises where a particular standard becomes outmoded. *See* Farrell, *supra* note 148, at 36-37. If the consumers who do not benefit directly from interoperability are excluded from this calculus, the perpetuation of outmoded standards is less likely to occur.

318. Lotus Development Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 79 (D. Mass. 1990).

319. Consumers of computer software do not actually own the software that they purchase. What they actually purchase is a software license for the copy which they possess.

320. *See* Lewis Galoob Toys v. Nintendo of Am., 780 F. Supp. 1283, 1290 (N.D. Cal. 1991) (citing Sony Corp. v. Universal City Studios, 464 U.S. 417, 429 (1984)); *see also* Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 261 (5th Cir. 1988) (refusing to find copyright infringement notwithstanding unintended use of copyrighted software).

321. 17 U.S.C. § 107 (1994) ("Notwithstanding the provisions of sections 106 and 106A, the fair use of a copyrighted work . . . is not an infringement of copyright.").

322. *See Manifesto, supra* note 7, at 2368; *see also* Farrell, *supra* note 148, at 36 (describing the benefits of interoperability standards in terms of "network externalities").

323. *See* Manifesto, *supra* note 7, at 2379 (arguing that the economic return justifies the investment in innovation); *see also* Farrell, *supra* note 148, at 36-37 (suggesting that the loss of incentive to innovate can result from excess standardization).

The district court opinion in *Lotus* made no accommodation for interoperability concerns. It would overprotect the Lotus menu, denying the program users the right to retain their personal investment in learning and developing macros. By contrast, the First Circuit opinion in *Lotus* allowed interoperability concerns to completely supersede established copyright principles. It would underprotect the Lotus menu, affording no protection to a valuable and expressive component of the 1-2-3 program. A balance must be struck between these two extremes. The value of the limited fair use approach lies in the recognition of both the benefits of interoperability to the public and the important role of copyright protection in ensuring the continuing creation of expressive works.

CONCLUSION

The proper method of evaluating the interoperability issues raised by copyrighted computer programs should take a market-oriented fair use approach. Courts must distinguish between uses that facilitate interoperability and uses that directly attack the market for the copyrighted work. Where a particular use of copyrighted material facilitates interoperability and does not result in unfair harm to the copyrighted work, the use should qualify as a fair use. By contrast, uses that facilitate interoperability but also carry a substantial risk of unfair harm to the copyrighted work cannot receive the court's approval if copyright protection is to be meaningful and effective.

The copying of user interface components can be a method of achieving interoperability as well as allowing innovative programs to compete on more equal footing with established predecessors. The permissibility of the use should depend on the critical distinction between primary and secondary markets. Permission for the use should therefore be predicated on proof of ownership of the original. If it is not, the goal of achieving interoperability can act as a rationalization for undermining the market for the original. Market destructive forces will arise from allowing interoperability goals to overwhelm all protection for important, expressive elements of computer programs. The law must recognize both the benefits and the limitations of interoperability to advance the arts and sciences effectively.