

PRIVILEGED USE: HAS JUDGE BOUDIN  
SUGGESTED A VIABLE MEANS OF  
COPYRIGHT PROTECTION FOR THE NON-  
LITERAL ASPECTS OF COMPUTER  
SOFTWARE IN *LOTUS DEVELOPMENT CORP.*  
*v. BORLAND INTERNATIONAL?*

DAVID M. MAIORANA\*

TABLE OF CONTENTS

Introduction .....	151
I. Development of the Relationship Among Computer Software, Copyright Law, and Competition .....	157
A. Computer Terminology Introduction .....	157
B. Introduction to Copyright Law .....	157
C. Computer Software Copyright Cases .....	158
1. <i>Whelan Associates, Inc. v. Jaslow Dental         Laboratory, Inc.</i> .....	159
2. <i>Lotus Development Corp. v. Paperback         Software International</i> .....	159
3. <i>Computer Associates International, Inc.         v. Altai, Inc.</i> .....	161
D. Computer Software Copyright Cases Involving Antitrust Law .....	162
1. <i>Lasercomb America, Inc. v. Reynolds</i> .....	163
2. <i>Corsearch, Inc. v. Thomson &amp; Thomson</i> .....	164

---

\* J.D. Candidate, May 1998, *The American University, Washington College of Law*. I would like to thank Professor Peter Jaszi for his valuable guidance and numerous suggestions on this paper. I also would like to thank my parents for their love and support. To my wife, Denise, thank you does not begin to describe the depth of my gratitude. Without your love and encouragement, this paper would not have been possible.

3.	<i>Data General Corp. v. Grumman Systems Support Corp.</i> . . . . .	165
E.	Monopolization Arguments For and Against Software Copyright . . . . .	166
1.	Arguments for limited copyright protection for software . . . . .	167
a.	Ex post monopoly ("locked in") . . . . .	168
b.	De facto monopoly . . . . .	168
c.	Uncertainty in the law—"chilling" innovation . . . . .	169
2.	Arguments for broader software copyright protection . . . . .	170
a.	Protection for the fruits of software developers' labors . . . . .	170
b.	Merger as a doctrine to prevent monopolization . . . . .	171
c.	Lead time is inadequate for computer software . . . . .	172
d.	Judge Keeton's arguments in <i>Paperback Software</i> . . . . .	172
II.	<i>Lotus Development Corp. v. Borland International</i> . . . . .	173
A.	The Majority Opinion . . . . .	173
B.	Concurrence by Judge Boudin . . . . .	175
1.	Judge Boudin's antitrust concerns . . . . .	175
2.	Judge Boudin suggests another solution—"privileged use" . . . . .	176
C.	Privileged Use Is Essentially a Limited Compulsory Licensing Scheme . . . . .	176
1.	Arguments for compulsory licensing . . . . .	177
2.	Arguments against compulsory licensing . . . . .	179
3.	Privileged use versus refusal to license . . . . .	180
4.	Privileged use is analogous to "fair use" . . . . .	180
III.	Privileged Use Is a Viable Means of Copyright Protection for Non-Literal Aspects of Computer Software . . . . .	182
A.	The Advantages of Privileged Use . . . . .	183
1.	Promoting standardization in the software industry . . . . .	183
2.	Encouraging software innovation . . . . .	183

3. Providing compensation to copyright owners . . . . . 183

4. Fostering competition . . . . . 184

5. Preventing monopolization by copyright owners . . . . . 184

B. What Copyright Standard Should Be Employed? . . 185

C. Implementation of Privileged Use . . . . . 186

Conclusion . . . . . 187

INTRODUCTION

The unique nature of computer software has made the application of traditional copyright law to programs a difficult task.<sup>1</sup> The difficulty arises because, although computer programs are considered literary works,<sup>2</sup> they differ from traditional copyrighted materials in that they have utility; they produce a result. Congress recognized the utility of computer programs when it amended the Copyright Act of 1976 to define explicitly the term "computer program."<sup>3</sup> In addition, computer software has both literal and non-literal aspects.<sup>4</sup> Although

1. See, e.g., *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807, 828 (1st Cir. 1995) (Boudin, J., concurring) ("Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit."), *aff'd by an equally divided court*, 116 S. Ct. 804 (1996) (per curiam); *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1341 (5th Cir. 1994) (noting that court decisions concerning copyrightability of computer software are in state of "creative ferment"); *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992) (noting that application of copyright law to computer software is "attempt to fit the proverbial square peg in a round hole"); Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329, 1329 (1987) (stating that computer software does not fit neatly into traditional intellectual property law); Joseph T. Verdesca, Jr., *Copyrighting the User Interface: Too Much Protection?*, 45 SW. L.J. 1047, 1079 (1991) (arguing that copyright law does not take into account unique nature of computer software).

2. See 17 U.S.C. § 102(a) (1994) (setting forth illustrative list of works that, if original, merit copyright protection). First on the list are "literary works," which are defined as "works, other than audiovisual works, expressed in words, numbers or other verbal or numerical symbols or indicia." *Id.* § 101. Literal aspects of computer software code traditionally have been considered literary works for purposes of copyright protection. See *infra* note 5 (discussing brief legislative history of characterization of computer programs as literary works).

3. See 17 U.S.C. § 101 (defining computer program as a set of statements or instructions to be used in a computer to bring about a certain result); see also *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 54-62 (D. Mass. 1990) (discussing thoroughly utilitarian nature of computer software and how it affects analysis of copyrightability of software); Aram Dobalian, *Copyright Protection for the Non-Literal Elements of Computer Programs: The Need for Compulsory Licensing*, 15 WHITTIER L. REV. 1019, 1019 (1994) (noting that, due to unusual nature of computer software, copyright may not be most appropriate area of law to apply).

4. See Menell, *supra* note 1, at 1334 (noting that literal aspects of computer software consist of program's "object code" and "source code"). The source code is what the program's creator types in a computer language that people can read and understand. See *id.* The source code is then translated into a form (binary language) that the computer can understand. See *id.*

The non-literal aspects of computer software include all portions of the program other than the source and object codes. See Matthew P. Larvick, *Questioning the Necessity of Copyright Protection for Software Interfaces*, 1994 U. ILL. L. REV. 187, 189. The non-literal aspects include the overall

Congress amended the Copyright Act of 1976 to include explicitly computer programs,<sup>5</sup> courts have struggled to determine the optimum level of copyright protection for computer software.<sup>6</sup> The amount of protection afforded to the non-literal aspects of computer software has varied in recent years from broad coverage to practically no coverage at all.<sup>7</sup>

Recognizing the problems involved in applying copyright law to computer software, several authors have proposed *sui generis* systems of protection for computer software.<sup>8</sup> These systems attempt to

organization of a program, the structure of a program's command system, the program's flow charts, and the screen presentation. See *Altai*, 982 F.2d at 702 (noting that non-literal components include flow charts, organization, parameter lists, and macros); *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989) (describing non-literal components of computer program as structure, sequence, organization, and user interface); *Paperback Software*, 740 F. Supp. at 42-44 (listing non-literal aspects of computer software as overall organization, structure of command system, and screen presentation).

This Comment focuses on the debate regarding the amount of copyright protection necessary for non-literal aspects of computer software. The literal aspects of computer software, the source and object codes, traditionally have been granted copyright protection. See *infra* note 6 (listing cases in which courts granted copyright protection to literal aspects of software).

The terms "software" and "program" have become synonymous in the computer industry. For purposes of this Comment, the terms are used interchangeably.

5. See 17 U.S.C. § 101. The Copyright Act of 1976 did not mention computer programs explicitly, but it was clear from legislative history that Congress intended computer programs to be included in the term "literary works." See H.R. REP. NO. 94-1476, at 54 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667 ("The term 'literary works' . . . includes . . . computer programs."). To clarify any ambiguity, however, in 1980 Congress amended the Copyright Act to include computer programs explicitly. See Pub. L. No. 96-517, 94 Stat. 3015, 3028 (1980) (amending 17 U.S.C. § 101).

6. Courts consistently have held that the literal aspects of computer software are copyrightable if original. See, e.g., *Altai*, 982 F.2d at 702 (noting that it is well settled that source and object codes are copyrightable); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1233 (3d Cir. 1986) (noting that it is well established that copyright law protects program's source and object codes); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983) (holding that copyright law covers source and object codes); *Paperback Software*, 740 F. Supp. at 45 (stating that literal manifestations of computer software are copyrightable).

Courts are split, however, on whether copyright law protects the non-literal aspects of computer software. This Comment addresses the copyrightability of the non-literal aspects. See *infra* Parts I.A-C (discussing issue of copyrightability of non-literal aspects of computer software).

7. See *Whelan*, 797 F.2d at 1239 (noting that copyright protection extends beyond literal aspects of computer program); *Paperback Software*, 740 F. Supp. at 77 (holding that copyright protection "clearly and unequivocally" should extend to original, non-literal aspects of computer software). *But see* *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807 (1st Cir. 1995) (holding that non-literal aspect of Lotus' program is uncopyrightable), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996) (*per curiam*); *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1442 (9th Cir. 1994) (stating that non-literal aspects are afforded "thin" protection from copyright infringement), *cert. denied*, 115 S. Ct. 1176 (1995).

8. See, e.g., Vance F. Brown, Comment, *The Incompatibility of Copyright and Computer Software*, 66 N.C.L. REV. 977, 1007 (1988) (advocating "marketplace system of protection for software" that does not rely on copyright law); Irwin R. Gross, *A New Framework for Software Protection: Distinguishing Between Interactive and Non-Interactive Aspects of Computer Programs*, 20 RUTGERS COMPUTER & TECH. L.J. 107, 179 (1994) (proposing "soft-right" system of protection for non-literal aspects of computer software); Virginia R. Lyons, Note, *Carrying Copyright Too Far: The Inadequacy of the Current System of Protection for Computer Programs*, HASTINGS COMM. & ENT. L.J.

balance the competing interests of the computer software industry against those of its consumers by formulating legal protection without the use of existing copyright law.<sup>9</sup>

Underlying the battle over software copyright protection is the question of possible anticompetitive behavior by computer program copyright owners. In several recent copyright cases, courts have applied an antitrust analysis when considering the behavior of the software copyright owners.<sup>10</sup> The courts in these cases did not consider the traditional factors used in antitrust determinations,<sup>11</sup> thereby reserving traditional antitrust analysis<sup>12</sup> for traditional antitrust situations.<sup>13</sup> Therefore, when a software copyright owner enforces its copyright in an "anticompetitive" manner such that it

81, 96 (1989) (suggesting new system of protection for computer software not based on copyright law); Menell, *supra* note 1, at 1371 (promoting system of legal protection for computer software that would last only short time); John C. Phillips, Note, *Sui Generis Intellectual Property Protection for Computer Software*, 60 GEO. WASH. L. REV. 997, 1032 (1992) (proposing *sui generis* legislative plan based on importance of computer software industry to national economy); Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2365 (1994) (advocating market-oriented approach to legal protection of computer software).

9. See *supra* note 8 (naming authors who advocate *sui generis* systems of protection for computer software).

10. See, e.g., *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1185-86 (1st Cir. 1994) (relying on antitrust principles to decide copyright infringement case); *MAI Sys. Corp. v. Peak Computer, Inc.*, 991 F.2d 511, 523, 26 U.S.P.Q. (BNA) 1458, 1464 (9th Cir. 1993) (upholding preliminary injunction preventing defendant Peak, a computer service organization, from using copyrighted software to service plaintiff's computers, effectively eliminating defendant as competitor in computer maintenance market); *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 978 (4th Cir. 1990) (analyzing defendant's conduct with eye toward its anticompetitive nature).

11. The elements of a traditional section 2 Sherman Act antitrust violation were discussed by the Supreme Court in *United States v. Grinnell Corp.*, 384 U.S. 563, 570-71 (1966):

The offense of monopoly under § 2 of the Sherman Act has two elements: (1) the possession of monopoly power in the relevant market; and (2) the willful acquisition or maintenance of that power as distinguished from growth or development as a consequence of a superior product, business acumen, or historic accident.

*Id.* Courts that have relied on a quasi-antitrust analysis have not examined these factors. See *infra* notes 93-94 and accompanying text. For a discussion of this misapplication of antitrust law, see Anthony L. Clapes, *Software, Copyright and Competition: The Use of Antitrust Theory to Undercut Copyright Protection for Computer Programs*, in *INTELLECTUAL PROPERTY ANTITRUST 1995*, at 553 (PLI Pats., Copyrights, Trademarks, and Literary Prop. Practice Course Handbook Series No. G4-3942, 1995).

12. An example of this type of situation, known as "tying," is one in which a seller of a product will sell the copyrighted product only if the buyer also buys a different product offered by the seller. See *Eastman Kodak Co. v. Image Technical Servs., Inc.*, 504 U.S. 451, 461 (1992) (defining "tying" arrangement as "an agreement by a party to sell one product but only on the condition that the buyer also purchases a different (or tied) product, or at least agrees that he will not purchase that product from any other supplier" (quoting *Northern Pac. R. Co. v. United States*, 356 U.S. 1, 5-6 (1958))).

13. See *Jefferson Parish Hosp. v. Hyde*, 466 U.S. 2, 9 (1984) (noting that "certain tying arrangements pose unacceptable risk of stifling competition and therefore are unreasonable 'per se'"); *Digidyne Corp. v. Data Gen. Corp.*, 734 F.2d 1336, 1338 (9th Cir. 1984) (stating that if tying arrangement is shown to restrain competition unreasonably it is illegal).

stifles competition, courts have relied on antitrust principles without performing the requisite analysis under antitrust law.<sup>14</sup> This Comment focuses on how courts have addressed the "anticompetitive" behavior of computer software copyright owners, and on how this analysis has affected the level of copyright protection courts afford computer software.

In *Lasercomb America, Inc. v. Reynolds*,<sup>15</sup> the Fourth Circuit prevented Lasercomb, the owner of a valid and enforceable copyright, from recovering damages from an infringer because the court believed that Lasercomb "use[d] its copyright in a manner adverse to the public policy embodied in copyright law."<sup>16</sup> This defense to copyright infringement is known as the "copyright misuse" doctrine.<sup>17</sup> Applying this doctrine, the court held that Lasercomb used its copyright to control competition through the use of anticompetition clauses in its copyright licensing agreements.<sup>18</sup>

Contrary to *Lasercomb*, however, other courts that have considered the anticompetitive behavior of copyright owners have held that the owners merely were enforcing their copyrights as they were entitled to do under the Copyright Act.<sup>19</sup> Thus, a tension exists between the right of the copyright owner to enforce a legally valid copyright and the right of the computer software industry to be protected from the anticompetitive practices of monopolistic industry leaders.

In addition to the threat of possible anticompetitive acts by copyright owners, several authors have expressed concern that granting copyrights for computer software confers monopoly status on the copyright recipients due to the nature of software and of the

---

14. See *infra* notes 93-94 and accompanying text (discussing criticisms of use of quasi-antitrust analysis in computer software copyright cases).

15. 911 F.2d 970 (4th Cir. 1990).

16. *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 978 (4th Cir. 1990).

17. See *id.* at 976 (finding that copyright misuse is valid defense to infringement). In *Lasercomb*, the Fourth Circuit did not analyze the requisite Sherman Act factors. See *supra* note 11 (delineating Sherman Act factors). The court in *Lasercomb* held that "a [copyright] misuse need not be a violation of antitrust law in order to comprise an equitable defense to an infringement action." *Lasercomb*, 911 F.2d at 978.

18. See *id.*; see also *infra* notes 98-103 and accompanying text (summarizing court's rejection of Lasercomb's use of its copyright agreement).

19. See *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1187 (1st Cir. 1994) (holding that plaintiff's unilateral refusal to license its copyright is presumptively valid business justification for any immediate harm to consumers); *Advanced Computer Servs. of Mich., Inc. v. MAI Sys. Corp.*, 845 F. Supp. 356, 370 (E.D. Va. 1994) (holding that plaintiff's selective licensing of its copyright is not copyright misuse, but rather is enforcement of its copyright that it legally is entitled to do); *Lucasarts Entertainment Co. v. Humongous Entertainment Co.*, 870 F. Supp. 285, 290 (N.D. Cal. 1993) (stating that copyright owner has "untrammelled right" to decide whether to license its copyright and to dictate terms of such license); see also *infra* notes 93-125 and accompanying text (reviewing other cases involving alleged copyright misuse).

software industry.<sup>20</sup> These authors contend that computer software should receive limited copyright protection to prevent monopolization by the first entrant into the market.<sup>21</sup> Other commentators argue, conversely, for broad software copyright protection.<sup>22</sup> Moreover, not only is there a conflict regarding what is the proper conduct of computer software copyright owners, but a conflict also exists regarding whether computer software copyrights should be issued at all. Thus, any solution to the problem of copyright protection for computer software must balance the interests of copyright owners against the concerns of the software industry.

In his concurrence in *Lotus Development Corp. v. Borland International*,<sup>23</sup> Judge Boudin suggests a new approach to copyright protection of computer software.<sup>24</sup> Boudin first echoed the sentiments of several commentators, stating that granting a copyright for the Lotus spreadsheet program would confer monopoly status on Lotus because copyright protection would prevent competitors from entering the market with similar, competing products.<sup>25</sup> Lotus had sued Borland after Borland allegedly copied Lotus' menu command hierarchy into its own competing spreadsheet program.<sup>26</sup> Judge Boudin argued that Borland's use of Lotus' program could be considered a "privileged use" because Borland was attempting to allow former Lotus users to utilize their knowledge of the Lotus program while enjoying the advantages of Borland's product.<sup>27</sup> In other words, Borland was not trying to take advantage of Lotus' advances but rather was attempting to achieve compatibility<sup>28</sup> between its product and

20. See *infra* notes 127-47 and accompanying text (describing concerns of some commentators that grant of computer software copyright confers monopoly status on copyright owner).

21. See *infra* notes 127-47 and accompanying text (discussing commentators' view that software should receive limited protection).

22. See *infra* notes 148-66 and accompanying text (noting arguments for stronger copyright protection for computer software to protect interests of software developers).

23. 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996) (per curiam).

24. See *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807, 821 (1st Cir. 1995) (Boudin, J., concurring) (suggesting that "privileged use doctrine" be applied to computer software), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996) (per curiam).

25. See *id.* (Boudin, J., concurring) (opining that granting copyright to non-literal aspects of Lotus' program will confer monopoly status to Lotus); see also *infra* notes 182-87 and accompanying text (discussing Judge Boudin's fear that granting copyrights to Lotus would lead to monopolization of market).

26. See *Lotus*, 49 F.3d at 809.

27. See *id.* at 821 (Boudin, J., concurring).

28. Compatibility is one of the main reasons cited in support of the argument that copyright protection for non-literal aspects should be limited. A computer user can transfer files between compatible programs but not between incompatible programs. Standardization occurs when all programs of a given category are compatible so that users can transfer files freely between competing programs. See Gerard J. Lewis, Jr., Comment, *Lotus Development Corp. v.*

Lotus'.<sup>29</sup> Thus, under Judge Boudin's proposed alternative, computer software copyright owners would be required to allow their competitors limited use of their products to ensure compatibility between competing software and to prevent monopolization of the software market.<sup>30</sup> In essence, Judge Boudin's proposal would form a type of limited compulsory licensing scheme.<sup>31</sup>

This Comment analyzes Judge Boudin's proposal for a new computer software copyright doctrine in light of the anticompetitive practices of copyright owners and the idea that the computer software copyright itself can create a monopoly.<sup>32</sup> Part I reviews some major computer software cases involving copyright and antitrust principles that have shaped the law leading to the First Circuit's decision in *Lotus* and outlines the traditional arguments for and against granting software copyrights.<sup>33</sup> Part II examines the *Lotus* decision and the "privileged use" scheme suggested by Judge Boudin in his concurrence and analyzes the potential benefits and disadvantages of such a system.<sup>34</sup> Part III argues that Judge Boudin's suggestion is a viable alternative for computer software copyright protection because it provides incentives for innovation, it fosters competition while preserving compatibility, and it prevents monopolization by the initial entrant into the market.<sup>35</sup> This Comment concludes that courts should take a long look at the "privileged use" idea and at the

---

Paperback Software International: *Broad Copyright Protection for User Interfaces Ignores the Software Industry's Trend Toward Standardization*, 52 U. PITT. L. REV. 689, 692 (1991) (advocating benefits of standardization). If the non-literal aspects (for example, screen presentation) of a new program are compatible with an established program, then users will be able to move to the new program (if it is considered better than the old program) without much additional training. See *id.* at 693. Advocates of standardization argue that strong copyright protection for non-literal aspects of computer software prevents standardization because competitors of a copyrighted program must make their programs sufficiently different to avoid copyright infringement liability. See *infra* notes 126-47 and accompanying text (describing arguments for standardization in computer software industry).

29. See Lewis, *supra* note 28, at 697.

30. See *infra* notes 188-93 and accompanying text (discussing Judge Boudin's alternative to copyrighting computer software).

31. See *Lotus*, 49 F.3d at 821 (Boudin, J., concurring) (describing plan for limited licensing of *Lotus*). For a discussion of the many arguments both for and against compulsory licensing, see *infra* notes 197-219 and accompanying text.

32. See *infra* notes 188-93 and accompanying text (examining Judge Boudin's proposal for new software copyright protection analysis).

33. See *infra* notes 45-166 and accompanying text (discussing cases involving software copyright protection and antitrust principles).

34. See *infra* notes 167-232 and accompanying text (examining Judge Boudin's proposal for "privileged use" scheme of software copyright protection).

35. See *infra* notes 233-65 and accompanying text (examining advantages of "privileged use" as mechanism for software copyright protection).



benefits it would confer as an alternative to the current computer software copyright system.<sup>36</sup>

## I. DEVELOPMENT OF THE RELATIONSHIP AMONG COMPUTER SOFTWARE, COPYRIGHT LAW, AND COMPETITION

### A. *Computer Terminology Introduction*

A computer program<sup>37</sup> consists of both literal and non-literal aspects.<sup>38</sup> The term "literal aspects" refers to the code that makes up the program.<sup>39</sup> There are two types of code: the source code and the object code.<sup>40</sup> The "source code" is what a computer programmer creates using a computer language that he or she understands.<sup>41</sup> The source code then is translated into "object code," a language that the computer understands.<sup>42</sup>

The non-literal aspects of a computer program are those portions of the program other than the source codes and object codes.<sup>43</sup> The non-literal aspects include the overall organization of a program, the structure of a program's on-screen command system, the program's flow charts, and the screen presentation.<sup>44</sup> This Comment focuses on the debate over the copyrightability of the non-literal aspects of computer software.

### B. *Introduction to Copyright Law*

Although the traditional basis for protection under American copyright law is found in the Constitution,<sup>45</sup> the Copyright Act of 1976, as amended, sets forth the subject matter protected under current copyright law.<sup>46</sup> The literal aspects of computer software traditionally have fallen within the "literary works" section of the

36. See *infra* Part III (advocating "privileged use" as viable alternative to current copyright protection of computer software).

37. As noted earlier, throughout this Comment, the terms "program" and "software" are used interchangeably.

38. See Larvick, *supra* note 4, at 189.

39. See *id.*

40. See *id.*

41. See *id.*

42. See *id.*

43. See *id.*

44. See *id.*; see also *supra* note 4 (discussing differences between literal and non-literal aspects of computer software).

45. See U.S. CONST. art. I, § 8, cl. 8. (granting Congress 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").

46. See 17 U.S.C. § 102 (1994) (setting forth list of protectable subject matter); see also *supra* notes 2-3, 5 (explaining relevant provisions of Copyright Act).

Copyright Act.<sup>47</sup> The law is unclear, however, with respect to copyright protection for non-literal aspects of computer software.

The traditional rule used to determine what portions of a work are copyrightable is known as the "idea/expression" dichotomy.<sup>48</sup> Copyright law does not protect ideas; it protects the expressions of ideas. Copyright protection is not, however, limitless. The doctrines of "merger" or *scènes à faire* can diminish the extent of protection the law provides for a particular work.<sup>49</sup> Both of these limiting principles have been applied to computer software.<sup>50</sup>

### C. Computer Software Copyright Cases

The dispute in the courts regarding the proper degree of copyright protection for computer programs centers on the non-literal aspects of the software.<sup>51</sup> To evaluate the proposal set forth by Judge Boudin, it is necessary to understand the evolution of software copyright protection. The following cases illustrate the development of copyright protection for the non-literal aspects of computer software.

---

47. See *supra* note 2 (listing works categorized as "literary works") and *supra* note 5 (noting legislative history of Copyright Act of 1976 that added computer software to literary works section).

48. See *Lotus Dev. Corp. v. Paperback Software*, 740 F. Supp. 37, 53-54 (D. Mass. 1990) (describing idea/expression doctrine). Copyright law protects the expression in a work but not the idea behind the expression. See *Baker v. Selden*, 101 U.S. 99, 107 (1879) (holding that copyright law protects description of accounting method but not idea of specific kind of accounting method). The courts have had a difficult time in applying the idea/expression dichotomy. See *Verdesca*, *supra* note 1, at 1051 n.50 (citing cases in which courts struggled to apply doctrine). The problem lies in defining the idea and then in differentiating between the idea and the expression of that idea. See *id.* at 1052.

49. According to the doctrine of merger, an expression is not copyrightable if it is one of a limited number of ways of expressing a particular idea. See *Paperback Software*, 740 F. Supp. at 58-59. However, if there are numerous other ways of expressing an idea, each expression is afforded copyright protection. See *id.*

*Scènes à faire* is related closely to merger. *Scènes à faire* refers to incidents, characters or settings which, because they are essential to the treatment of a given topic and are therefore likely to recur in later treatments of the same topic, are uncopyrightable. See *id.* at 59. For a discussion of merger and *scènes à faire*, see *Larvick*, *supra* note 4, at 191.

50. See, e.g., *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989) (noting that merger and *scènes à faire* apply to computer software); *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1021, 1041 (N.D. Cal. 1992) (holding that many features of Apple's program are uncopyrightable because item's expression merges with its idea), *aff'd in part, rev'd in part*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995); *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986) (describing *scènes à faire* in computer software copyright case).

51. See *supra* notes 6-7 and accompanying text (discussing consistent holdings that literal aspects of computer software are copyrightable and division in courts regarding non-literal aspects of software).

### 1. Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.

The Third Circuit was the first appellate court to confront squarely the issue of the copyrightability of the non-literal aspects of computer software in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*<sup>52</sup> Whelan had developed a computer program for Jaslow to manage and organize its dental laboratory records.<sup>53</sup> Later, Jaslow developed a similar program on its own.<sup>54</sup> Whelan alleged that Jaslow's program was based on its program and filed suit for copyright infringement.<sup>55</sup>

After noting that copyright protection extends to the literal aspects of the computer program,<sup>56</sup> the district court held that Jaslow's program did not copy Whelan's source code or object code.<sup>57</sup> The district court then examined the non-literal aspects of the software and found substantial similarity between the overall structure of the two programs,<sup>58</sup> which led it to hold that Jaslow had infringed Whelan's copyright.<sup>59</sup>

The Third Circuit, utilizing the idea/expression test,<sup>60</sup> concluded (1) that the purpose of the Whelan program was to assist in the operations of a dental laboratory; and (2) that copyright law protected the particular way in which the program expressed that purpose.<sup>61</sup> The Third Circuit affirmed the district court decision, holding for the first time that copyright law extends beyond the literal aspect of a program.<sup>62</sup>

### 2. Lotus Development Corp. v. Paperback Software International

In *Lotus Development Corp. v. Paperback Software International*,<sup>63</sup> Lotus brought suit against Paperback Software International alleging that Paperback's spreadsheet "VP Planner" infringed Lotus' spreadsheet "1-2-3."<sup>64</sup> As in *Whelan*, the issue in *Paperback Software* was the

52. 797 F.2d 1222 (3d Cir. 1986).

53. See *Whelan*, 797 F.2d at 1225-26.

54. See *id.* at 1226.

55. See *id.* at 1227.

56. See *supra* note 6 (discussing case law establishing copyrightability of literal aspects of computer programs).

57. See *Whelan*, 797 F.2d at 1233.

58. See *id.*

59. See *id.* at 1229.

60. The court in *Whelan* held that the purpose of a program is its idea; everything that is not a part of the program's purpose is an expression that is protectable by copyright law. See *id.* at 1238.

61. See *id.*

62. See *id.* at 1248.

63. 740 F. Supp. 37 (D. Mass. 1990).

64. See *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 42 (D. Mass. 1990).

copyrightability of the non-literal aspects of Lotus' program, namely its screen display.<sup>65</sup>

The court set forth a three step test to determine the copyrightability of the non-literal aspects of a computer program.<sup>66</sup> First, relying on the idea/expression distinction, a court must determine a program's idea and distinguish it from the program's expression of that idea.<sup>67</sup> Second, once the "idea" is identified, a court must analyze the elements of a program's expression to determine if they are essential to the expression of the idea or if they are one of only a few ways of expressing the idea.<sup>68</sup> Third, a court must decide whether the elements of expression that are essential to the idea are a "substantial" part of the copyrightable work.<sup>69</sup> In other words, to be copyrightable, a particular element of expression in a program must be essential to the program.<sup>70</sup>

Judge Keeton, writing for the majority in *Paperback Software*, found that both programs at issue embodied the idea of an electronic spreadsheet.<sup>71</sup> Next, Judge Keeton found that both the "rotated L" layout of the spreadsheet and the use of the "/" key to access the menu merged with the idea of a spreadsheet and were not copyrightable.<sup>72</sup> Judge Keeton did conclude, however, that Lotus' particular expression of a spreadsheet menu did not merge with the idea because it was merely one of a number of ways of expressing the idea of an electronic spreadsheet.<sup>73</sup> Finally, Judge Keeton decided that Lotus' menu command structure met the third step of his test because the menu structure was a unique feature that made "1-2-3" extremely popular.<sup>74</sup> Having found that "1-2-3" met all three parts of its test for copyrightability, the court thus held that Lotus' menu command structure was copyrightable.<sup>75</sup>

Judge Keeton held, as did the Third Circuit in *Whelan*, that a computer program embodies only one idea and that all other

---

65. See *id.* at 45-46.

66. See *id.* at 60-61.

67. See *id.*

68. See *id.* at 61. Thus, in the second step the court applies the merger and *scènes à faire* doctrines to the program to determine which aspects are copyrightable. See *supra* notes 49-50 and accompanying text (discussing application of merger and *scènes à faire* doctrines to copyright cases).

69. See *Paperback Software*, 740 F. Supp. at 61.

70. Judge Keeton noted that the third step is not necessarily an exclusively quantitative determination, but is qualitative as well. See *id.*

71. See *id.* at 65.

72. See *id.* at 66.

73. See *id.* at 67-68.

74. See *id.* at 68.

75. See *id.*

elements of a program are expressions of that idea.<sup>76</sup> The Second Circuit, however, rejected the notion that a program can consist of only one idea, and it formulated a new test for determining the copyrightability of computer software in *Computer Associates International, Inc. v. Altai, Inc.*<sup>77</sup> This new analysis results in significantly less copyright protection for computer software than the previous tests used in *Whelan* and *Paperback Software*.

### 3. Computer Associates International, Inc. v. Altai, Inc.

In *Computer Associates International, Inc. v. Altai, Inc.*, the Second Circuit rejected the test for copyrightability set forth in *Whelan* and formulated its own test.<sup>78</sup> At issue in *Altai* was a job scheduling program (ADAPTER) owned by Computer Associates ("CA").<sup>79</sup> Claude F. Arney III, an employee of CA, left the company for a position at Altai.<sup>80</sup> When he left, Arney took copies of ADAPTER with him and copied approximately thirty percent of it into a new program called OSCAR.<sup>81</sup> CA eventually learned that Altai had misappropriated parts of its ADAPTER program for Altai's OSCAR program,<sup>82</sup> and filed suit against Altai claiming copyright infringement.<sup>83</sup>

The Second Circuit discussed the *Whelan* test and rejected the notion that a computer program can embody only one idea.<sup>84</sup> Instead, the court found that a single program may contain several different ideas.<sup>85</sup> To determine what parts of the program are protectable, the program first must be broken down into its "constitu-

76. See *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1238 (3d Cir. 1986) (noting that purpose of *Whelan* program was to assist in daily operations of dental laboratory); *Paperback Software*, 740 F. Supp. at 65 (stating that "idea" embodied in Lotus program was that of electronic spreadsheet).

77. 982 F.2d 693, 705-06 (2d Cir. 1992).

78. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705-06 (2d Cir. 1992).

79. See *id.* at 698. A job scheduling program is designed to create a schedule for a mainframe computer, thereby telling the computer the order in which to perform certain tasks. See *id.*

80. See *id.* at 699.

81. See *id.* at 699-700.

82. See *id.* Altai was unaware that Arney was copying ADAPTER into the new program OSCAR. See *id.* When Altai discovered the copying, it immediately created a new version of OSCAR without using the code from ADAPTER. See *id.* This new version, OSCAR 3.5, was sent to all of Altai's customers as a replacement for the previous OSCAR version (OSCAR 3.4). See *id.* At trial, Altai conceded that it had copied ADAPTER into OSCAR 3.4 but not into OSCAR 3.5. See *id.* at 701. The trial court found that Altai did not infringe ADAPTER in writing OSCAR 3.5, and Computer Associates appealed this holding to the Second Circuit. See *id.*

83. See *id.*

84. See *id.* at 705-06.

85. See *id.* at 706.

ent structural parts," a step the court called "abstraction."<sup>86</sup> In the next step, "filtration," a court examines the constituent parts to sift out all unprotectable material.<sup>87</sup> In the final step, "comparison," a court compares any elements found to be protected in the filtration step to the structure of the allegedly infringing program.<sup>88</sup> If any elements of the infringing program are substantially similar to the protected elements of the original program, a finding of infringement is warranted.<sup>89</sup>

One commentator criticized the *Altai* test because, by breaking the program down into smaller parts, the court fails to see the proverbial big picture, the overall "look and feel" of the program.<sup>90</sup> In addition, the test has been criticized as being difficult to apply because identifying the individual elements of a computer program can be problematic.<sup>91</sup>

As the foregoing cases demonstrate, the level of copyright protection for the non-literal aspects of computer software has evolved quite rapidly during the past decade.<sup>92</sup> Although the issue of how much copyright protection to afford non-literal aspects has been much debated, courts only recently have confronted a new argument that copyright owners engaging in anticompetitive practices should lose copyright protection for their computer programs. The following section examines this argument and how courts have addressed it.

#### D. Computer Software Copyright Cases Involving Antitrust Law

Whereas the dispute in the copyright infringement cases discussed above centered on the degree of copyright protection afforded to non-literal aspects of computer software, another issue has emerged in a few recent cases. Defendants in copyright infringement actions have raised the "copyright misuse" defense to prevent recovery by copyright owners,<sup>93</sup> alleging that copyright owners have engaged in

---

86. See *id.*

87. See *id.* at 707-10. In this step, copyright limiting doctrines such as merger and *scènes à faire* are used to filter out the unprotectable elements of the program. See *id.*; see also *supra* notes 49-50 and accompanying text (discussing merger and *scènes à faire*).

88. See *Altai*, 982 F.2d at 710-11.

89. See *id.* at 706.

90. See Dobalian, *supra* note 3, at 1064 (noting that abstraction test creates risk of eliminating protection for combination of program's individual elements).

91. See *id.* at 1057.

92. See *supra* notes 51-91 and accompanying text (discussing varying levels of copyright protection for non-literal aspects of computer software).

93. See *supra* notes 10-14 and accompanying text (discussing lack of universal acceptance of copyright misuse defense); see also Clapes, *supra* note 11, at 559, 575 (arguing that judiciary's traditional emphasis on antitrust principles has been biased against intellectual property rights and should not be used as excuse for depriving rights of software copyright owners); Ramsey

anticompetitive licensing practices and therefore are not entitled to copyright protection.<sup>94</sup> Courts have differed in their treatment of the copyright misuse defense in software copyright cases.<sup>95</sup>

To understand how Judge Boudin's proposal relates to the anticompetition defense asserted by infringement defendants, it is necessary to understand how parties in computer software copyright infringement actions have utilized antitrust law.<sup>96</sup> The following cases illustrate how courts have balanced the anticompetition arguments of infringement defendants against the enforcement rights of plaintiff copyright owners.<sup>97</sup>

### 1. Lasercomb America, Inc. v. Reynolds

In *Lasercomb America, Inc. v. Reynolds*,<sup>98</sup> the Fourth Circuit confronted these competing considerations and sided with the alleged infringer.<sup>99</sup> Lasercomb licensed its computer assisted die-making software to Reynolds' employer, Holiday Steel Rule Corporation.<sup>100</sup> The licensing agreement contained a clause that forbade the licensee's developing or assisting in the development of any kind of computer assisted die-making software.<sup>101</sup> Defendants argued that this clause constituted "copyright misuse" because it effectively would stifle any competition in the area of computer assisted die-making software.<sup>102</sup> The Fourth Circuit agreed, holding that by including the anticompetitive clause in its licenses, Lasercomb was "using its

Hanna, *Misusing Antitrust: The Search for Functional Copyright Misuse Standards*, 46 STAN. L. REV. 401, 445 (1994) (opining that application of antitrust analysis in copyright infringement cases could deter creativity); Toshiko Takenaka, *Extending the New Patent Misuse Limitation to Copyright: Lasercomb America, Inc. v. Reynolds*, 5 SOFTWARE L.J. 739, 764 (1992) (questioning applicability of misuse defense to copyright cases). *But see* Note, *Clarifying the Copyright Misuse Defense: The Role of Antitrust Standards and First Amendment Values*, 104 HARV. L. REV. 1289, 1308 (1991) (advocating use of copyright misuse defense to prevent anticompetitive actions by copyright owners).

94. *See Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 977 (4th Cir. 1990) (applying copyright misuse defense to anticompetitive clauses in plaintiff's licensing agreements); Takenaka, *supra* note 93, at 760 (arguing that copyright misuse defense should apply to all licensing practices).

95. *See infra* notes 98-125 and accompanying text (examining various judicial interpretations of copyright misuse defense).

96. *See infra* notes 98-125 and accompanying text (examining cases interpreting antitrust law in computer software copyright context).

97. *See supra* note 19 (listing cases that upheld copyright owners' right to enforce their copyright).

98. 911 F.2d 970 (4th Cir. 1990).

99. *See Lasercomb*, 911 F.2d at 979 (holding that anticompetitive language in licensing agreement that attempted to control competition was misuse of copyright).

100. *See id.* at 971. The software at issue was a program that assisted in the mechanized creation of a steel rule die. *See id.*

101. *See id.* at 972-73.

102. *See id.* at 972.

copyright in a manner adverse to the public policy embodied in copyright law."<sup>103</sup>

The decision in *Lasercomb* meant that a copyright owner's right to license its copyright was limited and subject to judicial review under an antitrust-like analysis. Not all courts, however, have agreed with the reasoning in *Lasercomb*.<sup>104</sup> As the following cases demonstrate, some courts regard the copyright owner's right to enforce its copyright as paramount, even to the detriment of competition in the software industry.

## 2. Corsearch, Inc. v. Thomson & Thomson

The First Circuit addressed the antitrust implications of licensing by computer software copyright owners in *Corsearch, Inc. v. Thomson & Thomson*,<sup>105</sup> although the case was not an infringement action.<sup>106</sup> Thomson was a vendor of computer trademark database searching services.<sup>107</sup> Thomson licensed its database service to Corsearch, thereby allowing Corsearch to sell search outputs to its customers.<sup>108</sup> Corsearch later decided that it wanted to offer its own searching services, so it obtained a copy of Thomson's CD-ROM trademark database under false pretenses and used that copy to service its customers.<sup>109</sup> Thomson eventually discovered Corsearch's misappropriation of the database and terminated Corsearch's license to Thomson's on-line service.<sup>110</sup>

103. See *id.* at 978. Another question in the copyright misuse debate has been what level of scrutiny to apply to potentially anticompetitive behavior. Must the conduct of the copyright owner violate antitrust law or merely some lesser standard? See Takenaka, *supra* note 93, at 746-47 (noting two views of misuse doctrine, one based on antitrust law and one based on lesser standard). This question has been answered in different ways. See, e.g., *Lasercomb*, 911 F.2d at 978 (holding that copyright misuse need not violate antitrust law to be equitable defense to infringement action); Hanna, *supra* note 93, at 422 (arguing against use of antitrust law as guideline in determining existence of copyright misuse). But see *Corsearch, Inc. v. Thomson & Thomson*, 792 F. Supp. 305, 322-23 (S.D.N.Y. 1992) (applying antitrust analysis to analyze defendant's copyright misuse argument); Note, *supra* note 93, at 1302-03 (suggesting use of antitrust principles when hearing claims of copyright misuse); Roger Ara, Note, *Redefining Copyright Misuse*, 81 COLUM. L. REV. 1291, 1307-14 (1981) (recommending that courts judge all copyright misuse claims using antitrust principles).

104. See *infra* notes 105-25 and accompanying text (examining cases in which antitrust principles played subordinate role to rights of copyright enforcement).

105. 792 F. Supp. 305 (S.D.N.Y. 1992).

106. See *Corsearch*, 792 F. Supp. at 322-23.

107. See *id.* at 307-08. On-line trademark searching is used by people who are considering adopting a new trademark and who want to make sure that others are not using the trademark already. See *id.* at 307.

108. See *id.* at 311.

109. See *id.* at 314-15. By using the CD-ROM, Corsearch was able to avoid paying at least \$26,000 to Thomson for use of its on-line service. See *id.* at 315.

110. See *id.* at 316.



Corsearch then filed suit, alleging that Thomson's termination of the copyright license was an antitrust violation because of Thomson's monopoly in the trademark database searching industry.<sup>111</sup> The court rejected Corsearch's claim that the termination of the license constituted anticompetitive practices.<sup>112</sup> The court noted that, under copyright law, the owner of a copyright has a right to license its copyright and to terminate or limit the use of its property in any manner it deems appropriate.<sup>113</sup> The court further stated that Thomson's copyright did not prevent Corsearch from entering the trademark searching market; Corsearch was free to create its own database and to offer its searching services to the market.<sup>114</sup> Thomson's termination of the license, therefore, was not an antitrust violation.<sup>115</sup>

Contrary to the Fourth Circuit's conclusion in *Lasercomb*, the First Circuit's holding in *Corsearch* permitted restrictive licensing practices even to the detriment of the copyright owner's competitors.<sup>116</sup> The question whether a copyright owner could go so far as to refuse to license its software to any of its competitors, however, remained unresolved.

### 3. Data General Corp. v. Grumman Systems Support Corp.

In *Data General Corp. v. Grumman Systems Support Corp.*,<sup>117</sup> the First Circuit considered the issue of whether Data General's outright refusal to license its software to Grumman constituted an antitrust violation.<sup>118</sup> Data General was a computer manufacturer and retailer, and Grumman was a third party computer maintenance company.<sup>119</sup> Grumman claimed that Data General, by refusing to license its "ADEX" maintenance software to Grumman, was furthering its alleged monopoly in the computer maintenance market in violation of antitrust law.<sup>120</sup> In contrast to *Lasercomb* and *Corsearch*, cases in which the copyright owner's restrictive license was at issue, this case was concerned with the copyright owner's refusal to license the copyright at all.

---

111. *See id.*

112. *See id.* at 322-23.

113. *See id.*

114. *See id.* at 329.

115. *See id.* at 322.

116. *See id.* at 322-23 (citing *Fox Film Corp. v. Doyal*, 286 U.S. 123, 127-28, 130 (1932)).

117. 36 F.3d 1147 (1st Cir. 1994).

118. *See Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1152 (1st Cir. 1994).

119. *See id.*

120. *See id.*

Although the court noted that refusing to license a copyright might harm consumers initially by stifling competition,<sup>121</sup> the court found that the refusal to license a software copyright was “a *presumptively* valid business justification for any immediate harm to consumers.”<sup>122</sup> Thus, in balancing the competing interests of copyright and antitrust law,<sup>123</sup> the court determined that in this context the “economic incentives fueled by the Copyright Act” outweighed the pro-competition policies of antitrust law.<sup>124</sup>

This finding conflicts with *Lasercomb*, in which the Fourth Circuit, concerned about anticompetitive practices, decided that antitrust law limits a copyright owner’s right to enforce its copyright.<sup>125</sup> As the split between the First and Fourth Circuits demonstrates, courts disagree on the proper balance to be struck between the rights of copyright owners and the rights of their competitors regarding anticompetitive behavior.

#### E. Monopolization Arguments For and Against Software Copyright

Although some courts have limited the copyright owner’s right to enforce its copyright, some commentators would have courts go even

121. *See id.* at 1187. Data General argued that its refusal to license may actually benefit consumers by creating an incentive for competitors to create their own, better products. *See id.* at 1184-85. The court seemed to reject this argument by stating, “[B]y no means is a monopolist’s refusal to license a copyright entirely ‘pro-competitive’ . . . .” *Id.* at 1185. The court softened its stance, however, by noting that “some type of presumption [that refusal to license is a valid business practice] may nevertheless be appropriate.” *Id.*

122. *Id.* at 1187 (emphasis added). The court based its decision on the concern that imposing antitrust liability on a copyright owner for refusing to license would frustrate the objectives of the Copyright Act. *See id.* at 1185-86. In a footnote, the court noted that it did not hold that an antitrust plaintiff never could rebut this presumption because there may be *rare* cases in which imposing antitrust liability would not frustrate the Copyright Act’s objectives. *See id.* at 1187 n.64. The court did not give any guidance regarding the types of situations in which the presumption might be rebuttable.

123. The court noted that because the granting of a copyright gives the recipient a limited monopoly, copyright law can frustrate “the primary purpose of the antitrust laws—to preserve competition.” *Id.* at 1184-85 (quoting *SCM Corp. v. Xerox Corp.*, 645 F.2d 1195, 1203 (2d Cir. 1981)).

124. *Id.* at 1187. The court opined that “the Copyright Act tolerates behavior that may harm both consumers and competitors.” *Id.* at 1184-85. The court cited several cases that discussed the advantages of granting a copyright monopoly and found these advantages to outweigh the antitrust considerations. *See id.*; *see also* *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349 (1991) (stating that primary objective of Copyright Act is “[t]o promote the Progress of Science and useful Arts” (quoting U.S. CONST. art. I, § 8, cl. 2)); *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 429 (1984) (noting that granting of copyright monopoly motivates creative activity of authors and inventors by providing “special reward”); *Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156 (1975) (recognizing that immediate effect of copyright law is to reward creative labor by securing fair return for author).

125. *See supra* notes 98-104 and accompanying text (examining various arguments put forth in *Lasercomb* regarding application of antitrust principles to copyright misuse cases).

further.<sup>126</sup> These commentators argue that non-literal aspects of computer software should be granted minimal, if any, copyright protection because the granting of a copyright gives the recipient a "monopoly" over that particular portion of the software industry.<sup>127</sup> This argument parallels the concern of copyright infringers<sup>128</sup> that owners of software copyrights will acquire monopoly power in the software industry and eliminate competitors.<sup>129</sup> By granting only minimal copyright protection, commentators and copyright infringement defendants argue, competitors are free to make software compatible<sup>130</sup> with the copyrighted program, thereby ensuring that the copyright owner cannot gain a monopoly in the industry. In *Lotus*, Judge Boudin expressed these same concerns about monopolization but suggested an alternative way to create compatibility in the software industry.<sup>131</sup> To understand the motivation behind Judge Boudin's proposal, one must consider the monopolization arguments.

### 1. *Arguments for limited copyright protection for software*

The following arguments propose limited or no copyright protection for non-literal aspects of computer software because of possible monopolization of the software industry by the copyright recipient.<sup>132</sup> The following section outlines the three forms of this argument.

---

126. See *infra* notes 127-47 and accompanying text (discussing practical problems of copyright protection in terms of lack of standardization and software monopolization).

127. These arguments focus on the notion that the utilitarian nature of computer software, coupled with the "unusual nature" of the software industry, provides protection to a software developer such that strong copyright protection is not necessary for non-literal aspects of computer software. See Larvick, *supra* note 4, at 207-08.

128. See *supra* Parts I.C.1-3 (discussing computer software cases that examine copyrightability of non-literal aspects of programs, in light of post-copyright activities of copyright owners).

129. For simplicity, this Comment refers to this group of arguments as the "monopolization" arguments. Although the cases in Parts I.C.1-3 focus on post-copyright activity by copyright owners, and the arguments in this section focus on the effects of granting the copyright itself, the reasoning in both situations is based primarily on antitrust law.

130. Compatibility in the software industry commonly is referred to as "standardization." See Lewis, *supra* note 28, at 692-93 (describing benefits of standardization in computer software industry); Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1094 (1989) (noting that first entrant into market could reap benefits of standardization); Verdesca, *supra* note 1, at 1076 (arguing against copyright protection for non-literal aspects of computer software because protection would prevent standardization); see also *infra* notes 138-44 and accompanying text (describing commentators' concerns regarding "de facto" monopoly, reflecting desire for industry standardization).

131. See *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) (per curiam).

132. See Larvick, *supra* note 4, at 209-10.

a. *Ex post monopoly* ("locked in")

The term "ex post monopoly" refers to the situation in which a user of a copyrighted computer program is "locked in" to that program because competitors' products are incompatible with the copyrighted program.<sup>133</sup> Incompatibility occurs when strong copyright protection requires competitors to develop software so different from the copyrighted program that users of the copyrighted program cannot switch easily to the new program.<sup>134</sup> After investing a significant amount of time and training into learning the copyrighted program, the user often is unwilling and perhaps is unable to shift that investment to a competing program because of the incompatibility.<sup>135</sup> Because users will not change to another product and lose the time and effort they invested to learn the copyrighted program, the copyright owner obtains a quasi-monopoly. As a result, users effectively are "locked in" to the copyrighted program.<sup>136</sup> Limiting copyright protection for non-literal aspects of computer software can help prevent such a situation by fostering compatibility in the software industry, thereby enabling users to switch to a competitor's program without a large capital investment.<sup>137</sup>

b. *De facto monopoly*

A "de facto" monopoly exists when a software developer introduces an innovative product to the market.<sup>138</sup> Because copyright law protects the literal aspects of a computer program,<sup>139</sup> competitors

133. See Lewis, *supra* note 28, at 715 (noting that standardization benefits software users by avoiding costs associated with learning to use incompatible program); Verdesca, *supra* note 1, at 1075-76 (arguing that copyright protection should not extend to non-literal aspects of computer software because compatibility and standardization in computer industry would not be possible). But see *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 69 (D. Mass. 1990) (holding that desire for standardization cannot transcend authors' rights to limited monopoly in their works).

134. See Larvick, *supra* note 4, at 210.

135. See *id.*

136. See *id.*; see also *Lotus*, 49 F.3d at 821 (Boudin, J., concurring) (opining that granting copyright to Lotus' program will "lock" Lotus users into program); Lewis, *supra* note 28, at 715 (stating that standardization will lower training costs associated with switching to competing product); Verdesca, *supra* note 1, at 1075 (noting that without compatibility, software user is "locked in" to particular computer program).

137. See Larvick, *supra* note 4, at 211 (stating that standardization will lower business' costs for computer training of employees); Verdesca, *supra* note 1, at 1076 (noting that compatibility allows users to switch between programs without spending tremendous amounts of time, effort, and money).

138. See Larvick, *supra* note 4, at 206-07 (noting that developers enjoy degree of monopoly power when they first release new product).

139. See *supra* note 6 and accompanying text (describing how courts have held literal aspects of software copyrightable).

must write an entirely different literal code to avoid infringing the copyright of the newly released program.<sup>140</sup> Consequently, there is a time lag between the release of the first product and the release of competitors' products.<sup>141</sup> This time lag creates the de facto monopoly for the developer who releases its product first.<sup>142</sup> Commentators argue that copyrighting computer software augments de facto monopolies by giving copyright owners even greater control over the software industry, thereby enabling them to engage in monopoly pricing.<sup>143</sup> Commentators also fear that a de facto monopoly could give the copyright owner a monopoly over the idea behind the software, something that copyright law ordinarily does not protect.<sup>144</sup>

*c. Uncertainty in the law—"chilling" innovation*

Another argument against copyright protection for non-literal portions of software contends that granting copyright protection creates uncertainty in the software industry as to what competitors can and cannot copy.<sup>145</sup> In other words, the unsettled state of the law regarding copyright protection for non-literal aspects of computer software "chills" innovation because competitors are unwilling to make competing products out of fear of infringement suits.<sup>146</sup> Competi-

140. See Larvick, *supra* note 4, at 207.

141. See *id.* (noting that developer can exploit this monopoly position by improving its program, while competitors try to imitate it); Ralph S. Brown, *Design Protection: An Overview*, 34 UCLA L. REV. 1341, 1388 (1987) (noting that developers gain market advantage through head start on competitors). But see Hanna, *supra* note 93, at 414 (arguing that copyright protection alone will not ensure market success of product); Note, *supra* note 93, at 1299 (remarking that obtaining copyright does not confer economic power).

142. See Larvick, *supra* note 4, at 207. But see Samuelson et al., *supra* note 8, at 2367 (stating that natural lead time for first developer to release program may not suffice to protect developer's interests).

143. See Larvick, *supra* note 4, at 209-10 (expressing concern that developer owning a de facto monopoly could price many consumers out of market through monopoly pricing); Menell, *supra* note 1, at 1094 (same).

144. See Verdesca, *supra* note 1, at 1077. Note that copyright law protects only expression, not ideas behind the expression. See *supra* notes 45-50 and accompanying text (introducing legal basis of copyright law).

145. See Larvick, *supra* note 4, at 203-04 (noting concern of members of software industry that they are unsure of "what's OK and what will get [them] sued."); Verdesca, *supra* note 1, at 1078 (remarking that uncertainty in software industry will impede new innovation).

146. In addition to concern about "chilled" innovation, commentators also are concerned that strong copyright protection for non-literal aspects of computer software favors wealthy companies that own the software copyrights. See Larvick, *supra* note 4, at 204. Larvick argues that large companies have the most to lose if copyright protection for non-literal aspects is reduced. See *id.*

Menell, however, argues that copyright protection is more important for smaller, less established companies. See Menell, *supra* note 1, at 1070-71. Menell argues that without copyright protection, a software developer's profits would be eaten up by competitors who quickly would introduce competing products. See *id.* at 1070. Thus, smaller companies would

tors are discouraged from creating competing products; consequently, the copyright owner acquires a quasi-monopoly. Commentators argue that restricting copyright protection to only the literal aspects of computer software will allay programmers' concerns because non-literal infringement often is difficult to determine, whereas infringement of literal code is readily discernible.<sup>147</sup>

## 2. Arguments for broader software copyright protection

Other commentators and judges advocate strong copyright protection for the non-literal aspects of computer software. The following arguments are those conventionally given in response to the arguments against copyright protection for computer software.

### a. Protection for the fruits of software developers' labor

A traditional argument for copyright protection emphasizes that copyright law protects the fruits of the authors' work.<sup>148</sup> If software developers know their creations will be afforded minimal copyright protection, thus limiting the possible reward, they will be less willing to invest the time and money necessary to create new programs.<sup>149</sup>

not be able to enter the market without an assurance that their innovations would be protected from competition. See *id.* at 1070-71. This position was echoed in *Lotus Development Corp. v. Paperback Software International*, 740 F. Supp. 37, 75 (D. Mass. 1990). In *Paperback Software*, Lotus (a large, wealthy company) argued that elimination of copyright protection for non-literal aspects of computer software would have the greatest effect on small developers. See *id.* Although the first developer to market would have a head start, once a large well-known company came on the market with its clone, the small developers would lose the fruits of their innovations. See *id.*; see also *supra* notes 138-44 and accompanying text (discussing argument that first developer to market obtains an advantage from being first).

147. See Larvick, *supra* note 4, at 213 (proposing that copyright law should give minimal protection to non-literal aspects of computer program). But see *Paperback Software*, 740 F. Supp. at 73. Judge Keeton rejected the defendant's argument that because software developers would like to know what they may copy, there should be a bright line distinction between the literal and non-literal aspects of computer software. See *id.* Judge Keeton stated that a judgmental, evaluative standard is more appropriate, and that Congress could have created a bright-line rule, but chose not to do so. See *id.*

148. See William T. McGrath, *Copyright Protection for User Interfaces in the Nineties: Of Perilous Journeys on the Shoulders of Giants*, 4 SOFTWARE L.J. 597, 601 (1991) (arguing that narrow protection for non-literal aspects of computer software will result in sharing of fruits of program's popularity by competitors who did not contribute to creation of program); see also *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 432 (1984) (noting that primary effect of copyright law is to secure fair return on author's labor); *Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156 (1975) (recognizing that copyright seeks to protect author's creative labor); *Mazer v. Stein*, 347 U.S. 201, 219 (1954) (concluding that best way to encourage authors and inventors is to reward individual effort); *Paperback Software*, 740 F. Supp. at 75 (stating that ability of software developers to obtain rewards for their works depends on protection given by copyright law).

149. See *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1184 (1st Cir. 1994) (discussing how copyright law encourages innovation by permitting author to earn monopoly profits (citing *Sony*, 464 U.S. at 429)); *Paperback Software*, 740 F. Supp. at 79 (stating that one objective of copyright law is to give protection to expression, thus encouraging innovation);

The argument that innovation will be "chilled," therefore, supports both sides of the debate.<sup>150</sup>

*b. Merger as a doctrine to prevent monopolization*

Although several commentators contend that granting a copyright for computer software will create a monopoly for the recipient, copyright law has an inherent mechanism for preventing monopolization.<sup>151</sup> The merger doctrine is a copyright limitation that prevents a copyright owner from gaining a monopoly over a particular idea.<sup>152</sup> When there are a limited number of ways to express an idea, an expression of that idea cannot be copyrighted.<sup>153</sup> If, however, there are an unlimited number of ways of expressing an idea, then an expression of the idea is copyrightable.<sup>154</sup> Thus, if there are other ways of expressing an idea, granting a copyright for one expression does not prevent others from creating their own, unique expression of the same idea.<sup>155</sup>

Clapes, *supra* note 11, at 202-03 (arguing that non-literal aspects of computer software should be accorded strong copyright protection to foster innovation and progress); Larvick, *supra* note 4, at 206 (noting that software companies will be less inclined to innovate if their works are unprotected).

150. See *supra* notes 145-47 and accompanying text (discussing how strong copyright protection for non-literal aspects of computer software will chill innovation).

151. See *supra* notes 98-125 and accompanying text (discussing arguments that copyright protection for computer software will grant monopoly to copyright owner).

152. See *supra* notes 49-50 and accompanying text (discussing definition of merger and its application to computer software).

153. See *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967).

When the uncopyrightable subject matter is very narrow, so that the topic necessarily requires, if not only one form of expression, at best only a limited number, to permit copyrighting would mean that a party or parties, by copyrighting a mere handful of forms, could exhaust all possibility of future use of the substance. In such circumstances it does not seem accurate to say that any particular form of expression comes from the subject matter. However, it is necessary to say that the subject matter would be appropriated by permitting the copyrighting of its expression. We cannot recognize copyright as a game of chess in which the public can be checkmated.

*Id.* (citations omitted).

154. See *Paperback Software*, 740 F. Supp. at 59. Judge Keeton relied on this concept in his decision that the Lotus spreadsheet was copyrightable. See *id.* at 67-68. He noted that because there were an unlimited number of ways of expressing the idea of an electronic spreadsheet, the particular expression chosen by Lotus was copyrightable. See *id.*

155. The *Paperback Software* decision also stated that strong copyright protection will prevent programmers from borrowing and improving on ideas from other programmers as is done traditionally in the software industry. See *id.* at 77. This argument has come to be known as "On the Shoulders of Giants" ("OTSOG"), meaning that innovation only comes when programmers rely on the creations of those coming before them. See *id.* Judge Keeton rejected this argument as well, noting that programmers are free to borrow the idea of an electronic spreadsheet, as it is not protected by copyright law, but they cannot borrow Lotus' chosen expression of that idea. See *id.* at 78.

c. *Lead time is inadequate for computer software*

The argument that software developers who are first to market a program benefit from their "head start" has met with some criticism.<sup>156</sup> Professor Samuelson notes that the "lead time" contention is inapplicable to computer software because "information products, such as computer software, bear so much of the technical know-how required to make them on or near the surface of the product."<sup>157</sup> For this reason, computer software needs legal protection to safeguard the "know-how" contained therein.<sup>158</sup>

d. *Judge Keeton's arguments in Paperback Software*

In *Paperback Software*,<sup>159</sup> Judge Keeton addressed several of the above outlined arguments against copyright protection for computer software.<sup>160</sup> Judge Keeton was strongly opposed to the "de facto" monopoly argument,<sup>161</sup> which he summarily dismissed.<sup>162</sup> He opined that such an argument "flipped copyright on its head" and was "perverse."<sup>163</sup>

Judge Keeton also considered Paperback's argument in favor of a bright-line rule that would limit software copyright protection to only its literal aspects and thereby enable computer programmers to determine readily if they are infringing a copyrighted program.<sup>164</sup> He noted that the desire for certainty in the law is not unique to copyright law,<sup>165</sup> but that bright-line rules, because they permit little or no discretion, often fail to accommodate conflicting interests such as those present in *Paperback Software*.<sup>166</sup>

156. See *supra* notes 138-44 and accompanying text (discussing benefit of "lead time").

157. See Samuelson et al., *supra* note 8, at 2367 (explaining how traditional lead time is not applicable to computer software).

158. See *id.* at 2367-68; see also *Paperback Software*, 740 F. Supp. at 75 (noting that competitors can create a copy of a computer program "fairly promptly" once it is released on the market). This theory is in direct conflict with those of commentators who argue that strong copyright protection is not necessary for computer software because of the lead time that first-comers enjoy when entering the software market. See Larvick, *supra* note 4, at 206-07 (noting benefit of time lag between release of software product and competitors being able to clone product).

159. 740 F. Supp. 37 (D. Mass. 1990).

160. *Paperback Software*, 740 F. Supp. at 71-79.

161. See *id.* at 77-79.

162. See *id.* at 37, 79.

163. See *id.*

164. See *id.* at 73.

165. See *id.*

166. See *id.*



## II. *LOTUS DEVELOPMENT CORP. V. BORLAND INTERNATIONAL*

In addition to the ever-changing debate regarding copyright protection for non-literal aspects of software, the question concerning the interaction of antitrust law and copyright law remained unresolved when the First Circuit heard oral arguments on October 6, 1994, in *Lotus Development Corp. v. Borland International*.<sup>167</sup> The decision the court handed down on March 9, 1995, drastically changed the amount of copyright protection afforded to non-literal aspects of computer software.

### A. *The Majority Opinion*

The only issue in *Lotus* was whether the menu command hierarchy of Lotus' "1-2-3" spreadsheet was copyrightable.<sup>168</sup> The First Circuit began by rejecting the abstraction test set forth by the Second Circuit in *Altai*.<sup>169</sup> The First Circuit deemed the abstraction test "misleading," reasoning that the analysis encourages courts to find a basic level of copyrightable subject matter in a particular program.<sup>170</sup> The court further noted that the abstraction test "obscures" the more central question of whether Lotus' menu structure could be copyrighted at all.<sup>171</sup>

167. 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 807 (1996) (*per curiam*).

168. *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807, 813 (1st Cir. 1995). Lotus filed suit against Borland in the District of Massachusetts only four days after Judge Keeton's favorable opinion in *Paperback Software*. *See id.* at 810. Three days earlier, Borland had filed a declaratory action against Lotus in the Northern District of California, seeking a judgment of non-infringement. *See id.* The district court in California dismissed Borland's declaratory judgment action in favor of the suit in Massachusetts. *See id.* After numerous motions for summary judgment from both sides, Judge Keeton (the judge in *Paperback Software*) granted partial summary judgment for Lotus, holding that Lotus' menu structure was copyrightable. *See id.* On appeal, Borland conceded that Lotus' copyright was valid and that Borland had copied Lotus' menu structure. *See id.* at 813. Thus, the only remaining issue was whether the menu structure was copyrightable. *See id.*

169. *See supra* notes 78-91 and accompanying text (discussing reasons for replacing *Altai* test).

170. *See Lotus*, 49 F.3d at 815.

171. *See id.* The court seemed from the outset to have decided that Lotus' menu structure was not copyrightable. By dismissing the *Altai* test, the court ignored several cases (including *Altai* itself) that agreed that the purpose of the abstraction test *was* to determine copyrightability. *See, e.g.,* *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1343-44 (5th Cir. 1994) (adopting abstraction test); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 536 n.19 (5th Cir. 1994) (using abstraction test for non-literal aspects); *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 834 (10th Cir. 1993) (approving *Altai* abstraction test for determination of copyrightability of software menu); *Autoskill, Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1489-91 (10th Cir. 1993) (using *Altai* test for software); *Mitek Holdings, Inc. v. Arce Eng'g Co., Inc.*, 864 F. Supp. 1568, 1577-78 (S.D. Fla. 1994) (applying *Altai* test); *Cognotec Servs., Ltd. v. Morgan Guar. Trust Co. of N.Y.*, 862 F. Supp. 45, 49-51 (S.D.N.Y. 1994) (relying on abstraction test to decide copyrightability of foreign currency exchange computer program); *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. 616, 624 (N.D. Cal. 1993),

The court next turned to the question of whether Lotus' menu structure was copyrightable and decided that it was not.<sup>172</sup> In lieu of the abstraction test, the court held that the Lotus menu structure was a "method of operation" and thus not copyrightable under current copyright law.<sup>173</sup> Notably, the court ignored the congressional mandate that computer programs be considered "literary works."<sup>174</sup> The First Circuit's conclusion thus decidedly favors companies that make their living by cloning other, successful programs.

Apparently, the monopolization arguments presented in *amicus* briefs filed on behalf of Borland heavily swayed the First Circuit panel.<sup>175</sup> In particular, the court relied on the compatibility argument,<sup>176</sup> labeling "absurd" the notion that users would have to perform the same function differently in various computer programs if Lotus' menu were copyrightable.<sup>177</sup> Additionally, the court noted that users would be unable to transfer macros from one program to another.<sup>178</sup>

*aff'd*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995) (following abstraction test).

172. See *Lotus*, 49 F.3d at 815.

173. See *id.* (citing 17 U.S.C. § 102(b) (1994)).

174. See *supra* note 5 (discussing legislative history of Copyright Act leading to categorization of computer programs as literary works). Items that are considered useful articles are traditionally given less copyright protection than literary works. See also Anthony L. Clapes & Jennifer M. Daniels, *Lotus v. Borland: Nightmare on Milk Street?*, 12 COMPUTER LAW. 16, 18 (1995) (noting that *Lotus* decision reduced already minimal copyright protection for computer software).

Even those courts that have held computer software to be a useful article and thus entitled to less protection gave more copyright protection than did the court in *Lotus*. See *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006 (N.D. Cal. 1992), *aff'd*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995). In *Apple*, the district court dissected Apple's program into individual aspects and held that virtually all of them were unprotectable. See *id.* at 1027-47. It noted, similar to the court in *Lotus*, that "an article which has 'any intrinsic utilitarian function' can be denied copyright protection except to the extent that its artistic features can be identified separately and are capable of existing independently as a work of art." *Id.* at 1023 (quoting *Fabrica, Inc. v. El Dorado Corp.*, 697 F.2d 890, 893 (9th Cir. 1993)). The court then concluded that Apple's user interface was a useful article because it was intended to assist users in interacting with the computer. See *id.* The similarity between the reasoning in *Apple* (purpose of user interface is to help users access the computer) and *Lotus* (purpose of menu in Lotus' program is to give users access to the program) is striking. In both cases, the courts ignored the congressional mandate that computer programs are literary works and instead afforded computer software little or no copyright protection.

175. See Clapes, *supra* note 11, at 555.

176. See *supra* notes 126-47 and accompanying text (describing various monopolization arguments against copyright protection for computer software).

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

178. See *id.* A macro is a command given to a computer program that represents a sequence of operations to be performed by the computer. Many computer programs create or allow users to create macros that can be used only within the program in which they are created. A macro created in one program will not work in another computer program unless the programs are compatible. See *id.* at 809; Howard C. Anawalt, *Lotus Development Corp. v. Borland*

### B. Concurrence by Judge Boudin

Judge Boudin, in his concurring opinion, echoed the First Circuit's concern over the antitrust ramifications of according copyright protection to non-literal aspects of computer software.<sup>179</sup> He noted that the importance of the case, as well as his slightly different analysis of the problem, prompted him to write separately.<sup>180</sup> Lamenting that "applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit,"<sup>181</sup> Judge Boudin proffered a novel solution to the computer software copyright problem.

#### 1. Judge Boudin's antitrust concerns

Like the majority in *Lotus*, Judge Boudin was influenced considerably by the monopolization arguments. He stated that if Lotus were granted a copyright on its menu structure, users would be "locked in" to Lotus' program.<sup>182</sup> He further noted that Lotus' "1-2-3" spreadsheet represented the "de facto" standard in the spreadsheet industry.<sup>183</sup> Because Lotus already had reaped a substantial reward for being first,<sup>184</sup> Judge Boudin reasoned, its competitors should be allowed to enable Lotus customers to switch to their programs, assuming that the competitive products were superior to Lotus' products.<sup>185</sup> Accordingly, Judge Boudin concluded that the question was not whether Borland's claim should succeed, but "on what basis."<sup>186</sup> Instead of eliminating copyright protection for non-literal

---

International, 116 S. Ct. 804 (1996) Part One: Borland and the Blizzard of '96, 12 SANTA CLARA COMP. & HIGH TECH. L.J. 489, 490 (1996).

179. See *Lotus*, 49 F.3d at 819-20 (Boudin, J., concurring) (stating that misapplication results when copyright law is applied to computer programs because substance of programs is non-literal, and protection may limit public access).

180. See *id.* at 819 (Boudin, J., concurring).

181. *Id.* at 820 (Boudin, J., concurring).

182. See *id.* at 821 (Boudin, J., concurring).

183. See *id.* (Boudin, J., concurring).

184. See *supra* notes 138-44 and accompanying text (describing "lead time" argument).

185. See *Lotus*, 49 F.3d at 821 (Boudin, J., concurring). Judge Boudin noted that if Borland's product is not better than Lotus', Lotus customers will not switch to Borland. See *id.* This begs the question of whether Borland should be allowed to appropriate the successful aspects of Lotus' program, make a few changes, and then call it a "better" product. Without any legal protection for the Lotus program, second comers such as Borland will be able to do just that. In addition, the "lead time" argument will not apply in every case so that the justification for allowing competitors to appropriate aspects of successful software notably will be absent in many cases. See Samuelson et al., *supra* note 8, at 2367 (explaining why computer software innovation is different from traditional technical products).

186. *Lotus*, 49 F.3d at 821 (Boudin, J., concurring).

aspects of software as the majority did, Judge Boudin suggested an alternative.<sup>187</sup>

## 2. Judge Boudin suggests another solution—"privileged use"

Judge Boudin regarded Borland's use of Lotus' menu structure a "privileged use" because he believed that Borland sought to make its program compatible with Lotus' program.<sup>188</sup> In other words, Borland was trying merely to give Lotus users the ability to transfer the investment they made in learning Lotus' menu structure to Borland's program.<sup>189</sup> Judge Boudin explained that privileged use would differ significantly from the "method of operation" analysis.<sup>190</sup> Under the method of operation analysis, non-literal aspects are afforded no copyright protection, thereby enabling competitors to copy those aspects at will.<sup>191</sup> Under the privileged use doctrine, however, non-literal aspects are afforded copyright protection insofar as competitors simply cannot copy those aspects without improving upon them.<sup>192</sup> Thus, in order to avail itself of the privileged use exception, a competitor of a copyright owner would have to add something of its own to the copyrighted product so that the finder of fact considers its program "better" than the copyrighted product.<sup>193</sup>

### C. *Privileged Use Is Essentially a Limited Compulsory Licensing Scheme*

The privileged use scheme proposed by Judge Boudin bears significant resemblance to compulsory licensing. Compulsory licensing would require a copyright owner to license its copyright to competitors in exchange for some type of fixed royalty.<sup>194</sup> Similarly, under the privileged use doctrine, copyright owners would be required to allow competitors to copy some non-literal aspects of the

187. See *id.* (Boudin, J., concurring). Judge Boudin paid lip service to the majority's conclusion, calling it a "defensible position." *Id.*

188. See *id.* (Boudin, J., concurring).

189. See *id.* (Boudin, J., concurring). Judge Boudin was concerned with the idea of program users being locked into one program because competitors are prevented from creating compatible software due to copyright protection.

190. See *id.* (Boudin, J., concurring).

191. See *id.* (Boudin, J., concurring).

192. See *id.* (Boudin, J., concurring).

193. See *id.* (Boudin, J., concurring). Judge Boudin acknowledged that this requirement would add confusion to the test because the factfinder would have to determine whether a program was "better" than the copyrighted program it emulated. See *id.* (Boudin, J., concurring). He also noted that privileged use would entail "a host of administrative problems" and would make it more difficult for potential infringers to predict how courts decide their cases. *Id.* at 821-22 (Boudin, J., concurring).

194. See Dobalian, *supra* note 3, at 1066 (explaining how fear of music recording monopoly led to first compulsory licensing scheme).

copyrighted software so long as the competitor's motive was to create compatibility between its program and the copyrighted program. Although Judge Boudin is not the first to suggest a compulsory licensing scheme,<sup>195</sup> his notion of privileged use differs in that it would apply only when the licensee's motive was to ensure compatibility between its program and the licensor's.<sup>196</sup> To analyze the merits of Judge Boudin's privileged use doctrine, several arguments for and against compulsory licensing are set forth below.

### 1. *Arguments for compulsory licensing*

There are numerous arguments in favor of a compulsory licensing scheme for computer software copyrights. One commentator asserts that the primary rationale for imposing a compulsory licensing scheme on the software industry is the desire to transfer wealth from a competitor (who copied the copyrighted program to create compatibility) to the copyright owner.<sup>197</sup> Judge Boudin sought to accomplish the same goal with his privileged use suggestion.<sup>198</sup>

A compulsory licensing scheme also would reduce the cost for potential licensees to locate certain information such as the owner of the license, the availability of a license, and the potential cost of a license.<sup>199</sup> This would encourage licensees to seek out copyright owners, thus increasing the potential number of software licenses and

195. See, e.g., *id.* at 1067 (advocating use of compulsory licensing for computer software copyrights); Paul Durdik, *Reverse Engineering as a Fair Use Defense to Software Copyright Infringement*, 34 JURIMETRICS J. 451, 469 (1994) (suggesting compulsory licensing as alternative to enjoining reverse engineering); Lyons, *supra* note 8, at 96-97 (noting that compulsory licensing is one possible solution for problem of legal protection for computer software); Heather J. Meeker, *Multimedia and Copyright*, 20 RUTGERS COMPUTER & TECH. L.J. 375, 409-10 (1994) (supporting compulsory licensing for computer software copyrights); Menell, *supra* note 1, at 1065 (discussing benefits of compulsory licensing scheme); Mary L. Mills, *New Technology and the Limitations of Copyright Law: An Argument for Finding Alternatives to Copyright Legislation in an Era of Rapid Technological Change*, 65 CHI-KENT L. REV. 307, 331-32 (1989) (describing aspects of compulsory licensing scheme); Ralph Oman, *The Compulsory License Redux: Will it Survive in a Changing Marketplace?*, 5 CARDOZO ARTS & ENT. L.J. 37, 40-48 (1986) (debating pros and cons of compulsory licensing plan).

196. Those who advocate compulsory licensing schemes generally do not differentiate between instances when the copying is for compatibility purposes and when it is not. See Dobalian, *supra* note 3, at 1066-68 (comparing compulsory licensing to player piano rolls that allow pianos to play a musical composition automatically).

197. See Durdik, *supra* note 195, at 469. Despite this benefit of compulsory licensing, Durdik concludes that compulsory licensing is unnecessary in the computer software context. See *id.*

198. See *Lotus*, 49 F.3d at 819, 821 (Boudin, J., concurring).

199. See Meeker, *supra* note 195, at 409-10; see also Menell, *supra* note 1, at 1066-67 (arguing that compulsory licensing system would prevent needless spending by competitors to copy program that is industry standard).

reducing transaction costs between copyright owners and licensees.<sup>200</sup>

Another commentator contends that compulsory licensing would provide an acceptable reward for the efforts of programmers and would not unreasonably diminish benefits to consumers.<sup>201</sup> In addition, compulsory licensing may be necessary when balancing the interests of copyright owners and consumers becomes untenable.<sup>202</sup> When technology such as computer software makes absolute rights under copyright law infeasible, compulsory licensing may be the only solution.<sup>203</sup>

Furthermore, a compulsory licensing scheme would prevent anticompetitive practices by copyright owners.<sup>204</sup> Software copyright owners who have achieved a large presence in the market are unlikely to license their products to competitors and risk losing market share.<sup>205</sup> Thus, compulsory licensing would ensure that other companies are able to enter the market and compete with the copyright owner. Notably this argument conflicts with precedent, which holds that a copyright owner can refuse absolutely to license a copyright.<sup>206</sup> The competing interest of the copyright owners in enforcing their copyrights again clashes with the interests of consumers and of the software industry in fostering competition and innovation.<sup>207</sup>

200. See Dobalian, *supra* note 3, at 1068. Dobalian notes that a similar rationale motivated Congress to create compulsory licensing schemes for the cable television industry and for use of secondary cable transmissions. See *id.*; see also Menell, *supra* note 1, at 1065 (noting that compulsory licensing scheme would increase access to industry standard computer program while providing rewards to copyright owners).

201. See Dobalian, *supra* note 3, at 1067-68.

202. See Mills, *supra* note 195, at 330.

203. See *id.* at 332. In addition, compulsory licensing may become necessary when technology makes enforcement impossible. See *id.*

204. See Dobalian, *supra* note 3, at 1068 (stating that copyright owner has little incentive to license without compulsory licensing); Meeker, *supra* note 195, at 410 (noting that compulsory license abolishes copyright owner's most potent weapon—refusal to license); Menell, *supra* note 1, at 1365 (touting potential for compulsory licensing to restricting anticompetitive practices of copyright owners).

205. See *supra* text accompanying notes 133-36 (discussing benefits to copyright owners who do not license their software). When a copyright owner has a large share of the market, the legal protection of the copyright is an effective bar to others who wish to enter the market. See Dobalian, *supra* note 3, at 1068. In *Lotus Development Corp. v. Paperback Software International*, 740 F. Supp. 37 (D. Mass. 1990), Judge Keeton rejected this argument by noting that Microsoft's Excel spreadsheet program entered the market as a competitor to Lotus' "1-2-3" spreadsheet and has been commercially successful. See *id.* at 78. Excel's success, however, probably is more a result of Microsoft's strong economic position in the software industry before entering the spreadsheet market. See Dobalian, *supra* note 3, at 1068.

206. See *supra* notes 93-125 and accompanying text (reviewing cases that have held for and against copyright owners).

207. See *infra* notes 220-22 and accompanying text (discussing friction between refusal to license and "privileged use").

## 2. *Arguments against compulsory licensing*

Aside from the proposition that copyright owners are free to refuse to license their copyrights,<sup>208</sup> there are more fundamental criticisms of compulsory licensing that focus on the administration of such a system. For example, compulsory licensing would create an administrative burden for the agency in charge of regulating the scheme.<sup>209</sup> Deciding which software should be licensed and what fees to charge the licensees potentially could be complex.<sup>210</sup>

In addition, compulsory licensing would raise concerns over how and by whom such a system would be conducted.<sup>211</sup> Anxiety about "government control over the conditions of authorship" would result if a governmental agency regulated compulsory licensing of software.<sup>212</sup> If the system were run privately,<sup>213</sup> copyright owners would retain control over their work and would remain able to receive compensation for competitors' use of the copyrighted work.<sup>214</sup>

Finally, as recognized by Judge Boudin, a compulsory licensing system would be more difficult to administer than the current system.<sup>215</sup> Such a system would impose cost and delay<sup>216</sup> and would make the outcomes of cases difficult to predict,<sup>217</sup> a problem that deeply concerns the software industry.<sup>218</sup> Moreover, because the availability of privileged use depends on the licensee's motive, courts will face the potentially difficult task of determining whether the licensee intended to make its program compatible with the

208. See *supra* notes 105-25 and accompanying text (discussing copyright owners' liberty to limit use of licenses granted).

209. See Lyons, *supra* note 8, at 97.

210. See *id.*

211. See *infra* notes 253-65 and accompanying text (discussing implementation of privileged use system).

212. See Mills, *supra* note 195, at 333-34. To avoid any concern about government regulation of copyrights, Mills recommends using a private agency to oversee the compulsory licensing arrangement. See *id.* at 336.

213. See *infra* notes 257-65 and accompanying text (considering benefits of privately-run compulsory licensing system).

214. See Mills, *supra* note 195, at 333 (noting that owner's loss of control over work is more acceptable under private system).

215. See *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807, 821-22 (1st Cir. 1995) (Boudin, J., concurring) (remarking that privileged use would create its own problems), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996) (per curiam).

216. See *id.* (Boudin, J., concurring).

217. See *id.* at 822 (Boudin, J., concurring).

218. See *supra* notes 145-47 and accompanying text (discussing software developers' desire for certainty regarding what they can and cannot copy).

copyrighted program or whether it merely sought to copy the software's non-literal aspects without improving upon them.<sup>219</sup>

### 3. *Privileged use versus refusal to license*

As noted above, a fundamental conflict exists between a copyright owner's right under the Copyright Act to protect a copyright by refusing to license and the software industry's desire to prevent monopolization and to foster standardization.<sup>220</sup> A compulsory licensing scheme—even a limited one such as the system Judge Boudin envisions—clearly would run afoul of a copyright owner's right to refuse to license a copyright.<sup>221</sup> Privileged use would force copyright owners to license their copyrights on the non-literal aspects of their software whenever a competitor wanted to copy those aspects in the name of compatibility.<sup>222</sup> Although a copyright owner's right to protect a copyright is an important consideration, the benefits of standardization within the software industry are more compelling. Therefore, this Comment proposes that the courts adopt Judge Boudin's privileged use doctrine to create a limited compulsory licensing system for non-literal aspects of computer software only for the purpose of promoting compatibility only.

### 4. *Privileged use is analogous to "fair use"*

The idea of allowing a copyright owner's competitor a privileged use of copyrighted material resembles the traditional copyright exception known as "fair use."<sup>223</sup> Codified in the Copyright Act of 1976,<sup>224</sup> this exception permitted the copying of copyrighted materi-

219. See *Lotus*, 49 F.3d at 820-21 (Boudin, J., concurring) (inferring from Borland's introduction of unique user interface and adoption of Lotus' menu only "as a fall-back option" that Borland intended to attract old Lotus customers by ensuring compatibility).

220. See *supra* notes 126-47 and accompanying text (discussing how standardization and monopolization are concerns of software industry).

221. See *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1187 (1st Cir. 1994) (holding that copyright owner's absolute refusal to license its software copyright is "presumptively valid"). Two of the three judges on the panel in *Data General* also were on the panel in *Lotus*. In *Data General*, a copyright infringement case focusing more on antitrust law, the panel held that the refusal to license was valid even though it obviously was an anticompetitive practice, but in *Lotus*, a pure copyright infringement case, the panel was concerned with Lotus' ability to monopolize the software market by using its copyright. See *Lotus*, 49 F.3d at 818 (noting that primary goal of copyright law is to advance society, not to benefit developers). The court held that Lotus' spreadsheet menu was not copyrightable. See *id.* at 819. In *Data General*, actual anticompetitive practices were ruled valid, see 36 F.3d at 1187-88, whereas in *Lotus*, the possibility of a monopoly caused the court to deny valuable copyright protection. See 49 F.3d at 818. These cases were decided only six months apart.

222. See *Lotus*, 49 F.3d at 821 (Boudin, J., concurring).

223. See *id.* (Boudin, J., concurring) (recognizing similarity between privileged use and fair use).

224. 17 U.S.C. § 107 (1994) (codifying exception to exclusive copyright rights).



als under certain circumstances. Examples of fair use include the production of copies "for purposes such as criticism, comment, news reporting, teaching[,] . . . scholarship, [and] research."<sup>225</sup> A person who copies a copyrighted work for one of these purposes is not subject to suit under copyright law. In this respect, fair use is similar to privileged use. A competitor can copy portions of copyrighted computer software under the privileged use doctrine without incurring liability so long as the competitor copies only to ensure compatibility between its program and the copyrighted software and compensates the copyright owner accordingly.<sup>226</sup> Thus, under both the fair use and the privileged use exceptions, copying a copyrighted work is permitted for certain, limited purposes.

Fair use differs from privileged use, however, with regard to the analysis applied in determining whether a given instance of copyright infringement falls within the exception. Congress has instructed courts to examine four factors to decide if a particular use of a copyrighted work constitutes fair use:

- (1) [T]he 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.<sup>227</sup>

Some courts regard the first factor as dispositive, holding that commercial use is presumptively unfair.<sup>228</sup> Privileged use, in the context of the computer software industry, almost always would be for a commercial purpose and therefore would not qualify as traditional fair use. In his concurrence in *Lotus*, Judge Boudin acknowledged the presumption that the fair use exception is "unavailable" when the use is commercial in nature but noted that "'presumptively' does not mean 'always.'"<sup>229</sup> Moreover, courts have observed recently that the

225. *Id.*

226. See *supra* notes 188-93 and accompanying text (defining privileged use).

227. 17 U.S.C. § 107(1)-(4).

228. See *Sega Enters. Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392, 1398 (N.D. Cal.) (relying on presumption that commercial use is unfair), *modified*, 977 F.2d 1510, 1523 (9th Cir. 1992) (holding that presumption of unfairness was overcome); see also *Acuff-Rose Music, Inc. v. Campbell*, 972 F.2d 1429, 1436 (6th Cir. 1992) (holding that commercial use of copyrighted work is presumptively unfair), *rev'd*, 114 S. Ct. 1164, 1174 (1994) (holding that presumption does not end judicial analysis of issue).

229. *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) (per curiam).

commercial nature of a work is only one element to be considered in the fair use analysis and have held certain commercial uses permissible under the fair use doctrine.<sup>230</sup> Thus, the privileged use of a copyrighted computer program as contemplated by Judge Boudin possibly could be considered a "fair" use.

Despite the similarity between the fair use and privileged use exceptions, the fourth factor considered under the fair use analysis distinguishes the traditional exception from Judge Boudin's suggestion. The fourth factor addresses the extent to which the use affects the potential market for, or value of, the copyrighted work.<sup>231</sup> In the case of privileged use, the market for the copyrighted work is affected adversely because the ultimate purpose of the privileged use is to make a product competitive with the copyrighted product. Thus, Judge Boudin's notion of privileged use is more expansive than the traditional fair use doctrine.<sup>232</sup>

### III. PRIVILEGED USE IS A VIABLE MEANS OF COPYRIGHT PROTECTION FOR NON-LITERAL ASPECTS OF COMPUTER SOFTWARE

This Comment recommends adoption of the privileged use doctrine for non-literal aspects of computer software because it provides several advantages over the alternative of affording no copyright protection to non-literal features. A limited compulsory licensing scheme such as privileged use would: (1) promote standardization in the software industry; (2) provide software developers with incentives to innovate; (3) furnish compensation to copyright owners for the use of the non-literal aspects of their works; (4) foster competition in the software industry; and (5) prevent monopolization by copyright owners.<sup>233</sup>

---

230. See *Campbell*, 114 S. Ct. at 1170-71 (asserting that courts should examine all four statutory factors); *Sega*, 977 F.2d at 1522 (ruling that statutory factors are not exclusive of one another).

231. See 17 U.S.C. § 107(4).

232. In *Sega*, however, the Ninth Circuit held that Accolade's use of Sega's copyrighted computer program was a fair use. See *Sega*, 977 F.2d at 1527-28. Accolade broke down the computer program in Sega's "Genesis" video game in order to learn how to make its own video games compatible with Sega's system. See *id.* at 1514-15. The court held that although Accolade used Sega's program in order to become a competitor in the video game market, this was not a bar to a finding of fair use. See *id.* at 1522. Because Accolade intended to create compatibility between its games and Sega's system, its use of Sega's copyrighted program was a fair use. See *id.* at 1522-23.

Analogizing this reasoning to the privileged use situation, it is possible that privileged use, as contemplated by Judge Boudin, could be considered fair use. See *Triad Sys. Corp. v. Southeastern Express Co.*, 64 F.3d 1330, 1336 (9th Cir. 1995) (noting that use that provides public benefit by creating compatibility is fair use), *cert. denied*, 116 S. Ct. 1015 (1996).

233. See *infra* notes 234-47 and accompanying text (describing benefits of privileged use doctrine).

### A. *The Advantages of Privileged Use*

#### 1. *Promoting standardization in the software industry*

A standardized software industry would offer many benefits. The interest of software consumers in user-friendly programs would be recognized by creating conventional, well-known ways of using different computer software.<sup>234</sup> In addition, standardization would lower training costs. Consumers would be able to learn more software in a shorter time because they readily could transfer their knowledge between programs.<sup>235</sup> Standardization also might increase the overall number and variety of programs available to consumers.<sup>236</sup> It also would be easier for computer users to transfer files between different software and among other users.<sup>237</sup>

#### 2. *Encouraging software innovation*

Other benefits arise by limiting the availability of the privileged use doctrine to situations in which the potential licensee wants to create compatibility. With respect to software developers, standardization fosters innovation by allowing programmers to build on pre-existing standards when creating new software.<sup>238</sup> Developers would not have to start from scratch each time they wished to create a program and therefore would be more likely to develop new software.<sup>239</sup>

#### 3. *Providing compensation to copyright owners*

Although the privileged use doctrine would compel copyright owners to license their copyrights in certain situations, the owners would receive compensation for doing so.<sup>240</sup> Compulsory licensing systems are equitable because licensees gain access to the software only after compensating the licensor.<sup>241</sup> Privileged use thus strikes an acceptable balance between the needs of the software industry and

234. See Lewis, *supra* note 28, at 715.

235. See *id.*; see also Larvick, *supra* note 4, at 211 (noting that standardization would lower training and mobility costs).

236. See Lewis, *supra* note 28, at 715.

237. See *id.* at 693.

238. See *id.* at 716.

239. See *id.*; see also Larvick, *supra* note 4, at 211 (noting that standardization would lower development costs by allowing developers to build on existing standards).

240. See Dobalian, *supra* note 3, at 1067-68 (arguing that compulsory licensing system would reward software developers adequately); Menell, *supra* note 1, at 1365 (stating that flexible compulsory licensing scheme would ensure rewards for software developers).

241. See Mills, *supra* note 195, at 332.

those of the copyright owners by ensuring that copyright owners receive fair compensation for the use of their works.

#### 4. *Fostering competition*

Standardization resulting from privileged use also would foster competition in the software industry. Access to copyrighted standards would enable other companies to enter the software industry because the standard would serve as a starting point from which the new company could build.<sup>242</sup> Consequently, the cost of developing new products would decrease and companies with fewer resources would be able to enter the market.<sup>243</sup> Without standardization, entry into the software market is cost-prohibitive for many smaller companies, thus perpetuating larger companies' hold on the industry.<sup>244</sup>

#### 5. *Preventing monopolization by copyright owners*

By compelling copyright owners to license, privileged use would hinder copyright owners' efforts to monopolize the software industry.<sup>245</sup> Although privileged use obviously would prevent copyright owners from refusing to license their copyrights, owners still would enjoy a semblance of monopoly power by being the first to release a particular product.<sup>246</sup> Privileged use, however, would prevent software users from being "locked in" to a copyrighted product, thereby denying the copyright owner an "ex post facto" monopoly.<sup>247</sup> Finally, privileged use would eliminate some uncertainty that plagues members of the software industry. With license in hand, software developers will know to what extent they can copy the copyrighted program.

---

242. See Lewis, *supra* note 28, at 716.

243. See *id.*

244. See *id.*

245. See Dobalian, *supra* note 3, at 1068 (noting that without compulsory licensing, copyright owners have little incentive to license use of non-literal aspects of their software); Cole M. Fauver, *Compulsory Patent Licensing in the United States: An Idea Whose Time Has Come*, 8 NW. J. INT'L L. & BUS. 666, 677 (1988) (arguing that compulsory licensing would reduce licensor's enjoyment of monopoly profits); Lyons, *supra* note 8, at 96-97 (stating that compulsory licensing would eliminate monopolies on non-literal aspects of computer software).

246. See Larvick, *supra* note 4, at 209 (stating that first software developer to reach market enjoys lead time over copiers). *But see* Samuelson et al., *supra* note 8, at 2367 (discrediting lead time argument as applied to computer software industry because competitors can easily analyze software data and copy it). For copyright owners to enjoy this lead time fully, the privileged use scheme probably would have to include some type of time delay before licensees could begin to apply for licenses.

247. See *supra* notes 133-37 and accompanying text (discussing problems associated with ex post facto monopolies).

### B. *What Copyright Standard Should Be Employed?*

Adopting a limited compulsory licensing standard does not, of course, answer the question of how to determine what should be copyrighted in the first instance. To answer this question, one should look to the existing analysis set forth in *Paperback Software*.<sup>248</sup>

Although *Paperback Software* has been criticized for granting too much protection to the non-literal aspects of computer software,<sup>249</sup> much of this criticism is premised on the notion that strong copyright protection creates a monopoly for the copyright owner.<sup>250</sup> Critics' fears of monopolization in the software industry are moot if one considers the *Paperback Software* standard in light of a compulsory licensing scheme such as privileged use.<sup>251</sup> Strong copyright protection becomes acceptable in conjunction with the privileged use doctrine because the software market will have access to non-literal aspects of the copyrighted program through compulsory licensing. If, on the other hand, the copyrightability standard is weak, the copyright owner will have little protection in circumstances in which privileged use is unavailable; that is, in those instances in which the copier does not seek compatibility, but wishes merely to exploit the non-literal aspects of the copyright owner's software. Strong copyright protection would protect the copyright owner against competitors who want to copy the program's non-literal aspects for reasons other than ensuring compatibility, and the privileged use doctrine would ensure that the copyright owner is compensated when competitors copy the software's non-literal aspects for purposes of standardization. Thus, the proper standard for determining the copyrightability of software under a privileged use system is a test that provides broad copyright protection for non-literal aspects, such as the test set forth in *Paperback Software*.<sup>252</sup>

---

248. See *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996) (per curiam); see also *supra* notes 63-77 and accompanying text (describing three-step test used to determine copyrightability of non-literal aspects of computer program).

249. See, e.g., *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807, 815 (1st Cir. 1995) (recognizing limitations of *Altai* test), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996) (per curiam); Dobalian, *supra* note 3, at 1063-64 (criticizing *Paperback Software* and *Altai* tests); Larvick, *supra* note 4, at 194 (noting dissatisfaction with test set forth in *Paperback Software*); Lewis, *supra* note 28, at 699-701 (reporting judicial rejection of *Paperback Software* standard).

250. See *supra* notes 126-47 and accompanying text (describing arguments against strong copyright protection based on monopolization concerns).

251. See *supra* notes 188-219 and accompanying text (considering privileged use as an alternative to elimination of copyright protection).

252. See *Paperback Software*, 740 F. Supp. at 59-62 (establishing three-step test to determine copyrightability of non-literal aspects of computer software).

### C. *Implementation of Privileged Use*

Before lawmakers implement a system like privileged use, they must consider how such a system would work. Congress could intervene and create a compulsory licensing system based on compatibility. Such systems are not unprecedented. In the Copyright Act of 1976, Congress created a compulsory licensing system for jukeboxes.<sup>253</sup> Jukebox operators were required to pay a fee, set forth in the statute, in order to play the music contained in their jukeboxes.<sup>254</sup>

Congress could legislate a similar system for computer software to implement privileged use. A software developer wishing to create a program compatible with a copyrighted program would be required to pay a fee to the copyright holder. Once the developer paid the fee, he or she could borrow the non-literal aspects of the copyrighted program to write a new program. Under a government-run system, a government entity would set the appropriate fees and royalties.<sup>255</sup> A privileged use system administered by the government, however, would raise concerns about excessive "government control over the conditions of authorship."<sup>256</sup>

An alternative to a government-run system would be a "market" system in which private parties address copyright infringement problems.<sup>257</sup> A privately-run system would keep pace better with the rapid technological advances that characterize the computer software industry.<sup>258</sup> Privately-run, market-based, compulsory licensing systems, however, have been criticized as inadequate.<sup>259</sup>

One pragmatic approach is a voluntary licensing system in which copyright owners could bargain for their own royalties with potential licensees.<sup>260</sup> Voluntary licensing, however, also has been criticized because it would be difficult for a copyright owner to negotiate with every potential licensee on an individual basis.<sup>261</sup> One commentator

253. See 17 U.S.C. § 116 (1994).

254. See *id.* § 116(b)(1); see also Oman, *supra* note 195, at 41 (noting that noncompliance with statute is continuing problem).

255. The Copyright Royalty Tribunal, created in the Copyright Act of 1976, controls compulsory licensing royalties in the United States. See *id.* § 801(a) (authorizing Librarian of Congress to establish copyright royalty panels).

256. See Mills, *supra* note 195, at 333-34 (reporting concern over potentially intrusive oversight and possible loss of, or sanction against, creative works).

257. See *id.* at 334.

258. See *id.* The fact that the system would be run by members of the industry would ensure that the licensing entity is up to date on any technological changes affecting the market.

259. See *id.* at 335. Mills notes that market-run systems are prone to unpredictability and thus are less attractive than government-run systems. See *id.* at 335-36.

260. See *id.* at 336.

261. See *id.*

has addressed this criticism by recommending that a private agency administer such a voluntary system.<sup>262</sup> The agency would perform the same function as the government entity in a government-run system without the concern of excessive government involvement.<sup>263</sup> Although the copyright owner would be free either to contract with the private agency or to negotiate on an individual basis,<sup>264</sup> the copyright owner would have to permit privileged use by competitors interested in creating compatible programs.<sup>265</sup>

Of course, Congress could institute a privileged use system that would not compensate copyright owners at all. It could decide that standardization in the computer software industry is so important that copyright owners must allow competitors to duplicate non-literal aspects of copyrighted software without receiving any fee or royalty. Copyright owners obviously would oppose this type of system. Therefore, a privileged use system would have to incorporate a compensation scheme. Of the possible alternatives, a voluntary licensing system in which copyright owners either would contract with a private licensing agency or would negotiate with individual licensees is the most viable because it avoids government control of works of authorship and it enables authors to determine the conditions under which their works are licensed.

#### CONCLUSION

The debate regarding the degree of copyright protection appropriate for non-literal aspects of computer software continues. Much of the dispute focuses on the problem of monopolization of the software market through anticompetitive practices by software copyright owners. To solve this problem, many commentators, as well as the First Circuit in *Lotus*, advocate diminishing (or eliminating altogether) copyright protection for non-literal features of software so that copyright owners cannot use copyrights to stifle or eliminate competition.

Judge Boudin, in his concurrence in *Lotus*, has suggested permitting a privileged use of the non-literal aspects of a copyright owner's software by a competitor seeking to make its program compatible with the copyrighted software. The competitor would receive a license to use the non-literal portions of the program and, in exchange, would

---

262. See *id.* at 336-37.

263. See *id.* at 337.

264. See *id.*

265. See *supra* notes 188-93 and accompanying text (defining privileged use).

pay a royalty to the copyright owner. Privileged use thus is a type of compulsory licensing scheme, limited insofar as it is available only when the licensee seeks compatibility between his or her program and the copyrighted program.

A limited compulsory licensing scheme such as privileged use would create standardization in the software industry, provide incentives to software developers to innovate, furnish compensation to copyright owners for use of the non-literal aspects of their works, foster competition in the software industry, and prevent monopolization by copyright owners.

Furthermore, the privileged use doctrine, coupled with strong copyright protection for non-literal aspects of computer software, would strike an acceptable balance between the interests of copyright owners and the software industry: copyright owners would receive compensation and competitors would obtain access to copyrighted software. This arrangement will serve to benefit software consumers in the future.