

Summer 1992

## The Interdependent Nature of Computer Software: Another Reason Why User-Interfaces Should Not Be Protected by Copyright Law, 25 J. Marshall L. Rev. 737 (1992)

David B. Fonda

Follow this and additional works at: <https://repository.law.uic.edu/lawreview>



Part of the [Computer Law Commons](#), and the [Intellectual Property Law Commons](#)

---

### Recommended Citation

David B. Fonda, The Interdependent Nature of Computer Software: Another Reason Why User-Interfaces Should Not Be Protected by Copyright Law, 25 J. Marshall L. Rev. 737 (1992)

<https://repository.law.uic.edu/lawreview/vol25/iss4/3>

This Article is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in UIC Law Review by an authorized administrator of UIC Law Open Access Repository. For more information, please contact [repository@jmls.edu](mailto:repository@jmls.edu).

# THE INTERDEPENDENT NATURE OF COMPUTER SOFTWARE: ANOTHER REASON WHY USER-INTERFACES SHOULD NOT BE PROTECTED BY COPYRIGHT LAW

DAVID B. FONDA\*

## INTRODUCTION

The first computers were developed in the 1940's.<sup>1</sup> In the 1950's and 1960's, computer programs automated electric switches that once were operated by hand.<sup>2</sup> With the advent of semiconductor technology in the 1970's, microcomputers became affordable to small businesses and to home users.<sup>3</sup> This new affordability opened new markets that led to remarkable advances in computer science throughout the 1980's. With every advance in computer hardware scores of software producers emerged, seeking to capitalize on the high demand for programs that would run on these new affordable computers.<sup>4</sup> The marketability of these software programs depended in large part on how easy they were for the typical user to operate.<sup>5</sup> The user-interface<sup>6</sup> of a program became an increasingly valuable part of the software package.<sup>7</sup> As the time, and consequently the cost, of producing more innovative user-interfaces increased, so did the amount of copying.<sup>8</sup> Throughout this information revolution, the courts "often struggled awkwardly to

---

\* David Fonda received a B.S. in Mathematics from Brigham Young University, and a J.D. from the Gonzaga University School of Law where he was the Editor-in-Chief of the Gonzaga Law Review. Mr. Fonda has developed programs for the National Security Agency and is currently a Judicial Clerk for the Honorable Norman Jackson of the Utah Court of Appeals.

1. Vance F. Brown, *The Incompatibility of Copyright and Computer Software: An Economic Evaluation and a Proposal for a Marketplace Solution*, 66 N.C. L. REV. 977, 979 (1988).

2. *Id.*

3. *Id.* at 979-80.

4. *Id.* at 980.

5. Brett N. Dorny & Michael K. Friedland, *Copyrighting "Look and Feel" Manufacturers: Technologies v. CAMS*, 3 HARV. J.L. & TECH. 195, 195 (1990).

6. The user-interface of the program is that interactive portion generated by the program that helps the user utilize the program. It may consist of help screens, pull-down menus, or various command sequences.

7. Dorny & Friedland, *supra* note 5, at 196.

8. *Id.*

provide adequate legal protection for the ever-changing industry."<sup>9</sup> Today, courts predominantly use copyright law to protect computer software.

The Constitution grants Congress the "power to promote the progress of science . . . by securing for limited times to authors . . . the exclusive right to their respective writings . . ."<sup>10</sup> Congress first exercised this power when it enacted Title 17 of the United States Code, the Copyright Act. Thus, the Constitutional purpose of the Copyright Act is to "promote the progress of science."<sup>11</sup> Since that first enactment, the Copyright Act has undergone significant change. Today, the Copyright Act<sup>12</sup> allows the "respective writings" language of the Constitution to include computer programs, and the "authors" language to include computer programmers.<sup>13</sup> The Copyright Act provides a definition of "computer program."<sup>14</sup> However, the definition speaks only of statements and instructions or "literal manifestations" of a program. Whether or not "nonliteral" manifestations, such as the user-interface of a program fall within the scope of the Copyright Act is left to judicial interpretation.<sup>15</sup> This article first explains the difference between literal and nonliteral manifestations of a program and defines user-interface. This article then analyzes how the judiciary presently interprets Section 102 of the Copyright Act<sup>16</sup> to grant copyright protection to user-interfaces. Finally, this article discusses how the interdependent nature of computer programs makes protecting the user-interface of a program through section 102 impossible and at odds with the constitutional purpose of the Copyright Act.

## I. THE LITERAL AND NONLITERAL MANIFESTATIONS OF COMPUTER SOFTWARE

Computer software has both literal and nonliteral manifestations of its existence. Particular aspects of software can be repro-

---

9. Keith A. Styracula, *The Adequacy of Copyright Protection for Computer Software in the European Community 1992: A Critical Analysis of the EC's Draft Directive*, 31 JURIMETRICS J. 329, 329 n.2 (1991).

10. U.S. CONST. art. I, § 8, cl. 8.

11. *Id.*

12. The Copyright Act that governs today is the 1976 Copyright Act, 17 U.S.C. §§ 100-914 (1988).

13. 17 U.S.C. § 101 (1988).

14. *Id.*

15. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 46 (D. Mass. 1990). In late April of this year, U.S. District Judge Vaughn Walker handed down an oral ruling in Apple Computer's case against Microsoft Corp., declaring that certain characteristics of a visual display are not protected by copyright law. Heather Clancy, *Ruling Could Spur Software Development—Apple Takes Hit in Microsoft*, COMPUTER RESELLER NEWS, Apr. 20, 1992, at 2.

16. 17 U.S.C. § 102 (1988).

duced in a tangible form, while others cannot. To fully understand the difference, it is helpful to consider what constitutes a particular piece of computer software.

### A. Computer Software Background

Computers only perform the tasks which programmers instruct them to perform. If they have no instructions to follow, they are essentially useless.<sup>17</sup> The instructions that govern all the functions of a computer are called the computer's program, or software. There are two basic types of software: operating system software and applications software.<sup>18</sup> Operating systems software such as DOS, UNIX, ZENIX, or OS/2 control the basic functions of the computer's hardware.<sup>19</sup> Applications software allows the user to perform a certain task such as word processing, spreadsheet calculations or database management.

The first step in software development is determining what the program must accomplish. Once the task is defined, software designers often create a flowchart outlining the exact steps that the program is to follow. A flowchart is a kind of symbolic outline or schematic representation of a computer program's logic. It is the first manifestation of the program.<sup>20</sup> The next step for the designer is to translate each logical step of the flowchart into a language or code that the computer can understand. Programming languages are either "low-level," "intermediate-level" or "high-level" depending on how closely the language corresponds to the language of the individual computer. The lowest level of programming language is called machine code.<sup>21</sup> Machine code is usually written in either a binary<sup>22</sup> or a hexadecimal form.<sup>23</sup> Machine code is specific to each computer and must be written in the machine code of the computer that on which the software will be executed. Software designers write software in machine code because software executes the fastest when written in the lowest level of programming language.<sup>24</sup> Software written in either an intermediate-level or a high-level programming language must be converted into machine language

---

17. ROBIN BRADBEER, ET AL., *THE BEGINNER'S GUIDE TO COMPUTERS* 8 (1982).

18. *Id.*

19. *Id.*

20. See, e.g., BRADBEER ET AL., *supra* note 17, at 59-61; Stephen Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 HARV. L. REV. 281, 341 n.235 (1970).

21. BRADBEER ET AL., *supra* note 17.

22. Binary code is represented by the characters "0" and "1." *Id.*

23. Hexadecimal code is represented by the characters 0-9 and A-F. Whereas one "byte" is represented by eight digits in binary code, a "byte" can be represented by just two digits in hexadecimal code. *Id.* at 197.

24. *Id.* at 198.

before it can be executed on the computer. This process is referred to either as assembling or compiling, depending on the level of programming language in which the software is written.<sup>25</sup> Software designers use intermediate-level or high-level programming languages to write software because the languages are similar to everyday language and are easier for the designer to understand.<sup>26</sup> Higher-level languages are also not machine specific and can be used on different computers because they are translated into the machine code of the computer they are executing on at the time that the computer software is being executed.

Software encoded in machine language is called object code and can be executed directly by the computer's Central Processing Unit (CPU) without need for assembling or compiling. Software written in any higher-level language is called source code and must be translated into machine code before being executed. Today, most software designers write their programs in a higher-level language like FORTRAN (FORMula TRANslator), COBOL (COMMON Business Oriented Language), Pascal, BASIC (Beginner's All-purpose Symbolic Instruction Code), or the language most programmers use, "C." Since software written in these languages must be converted into machine language, most programs today have both a source code and an object code.

### *B. The Interdependent Nature of Computer Software*

Most commercial software is composed of three interdependent parts. The first is what this author calls the "basic form" of the program, the second is the structure or organization of the program, and the third is the user-interface of the program.

The basic form of the program performs all of the steps of the flowchart, and directs the computer to perform the basic tasks and objectives that inspired the programmer to initially write the program. The basic form of a program is functional, but not very efficient or marketable. An efficient program, one that will perform the desired tasks in the fewest amount of programmed steps, is a function of the program's structure and organization. The programmer will restructure and reorganize the basic form until the program runs at maximum efficiency. This is an essential element of marketable software. This efficient, basic form of the program is still in a form that only someone with a significant amount of computer acumen can use. Marketable software must be easy for the average user to understand.

---

25. BRADBEER, ET AL., *supra* note 17 at 195.

26. *Id.* at 96.

The user-interface of a program packages the efficient basic form in a way that permits the average user to interact or "interface" with the computer. Its sole purpose is to allow the user to run the basic form of the program. The software designer may write source code that will make certain screens appear, or certain menus appear on the screen that offer a variety of choices to guide the user through the program. The designer may also designate a certain key or sequence of key strokes that will generate helpful hints which appear on the screen. Without the basic form of the program, there would be no need to write a user-interface portion of the program. However, the quality of the user-interface usually determines how well the overall program sells.

The last step in software development is preparing the documentation or instructions on how to install and use the program.

### *C. Literal and Nonliteral Manifestations of the Software*

The literal manifestations of a computer program are those that can be reproduced in a tangible form. The documentation of the software that is supplied to the user is clearly tangible and a literal manifestation of the software. The source and object code of a program can also be printed and put in a tangible form. They too are literal manifestations of the software. Even the flowchart, which is symbolic in nature can be put in a tangible form and is a partially literal manifestation of the software. The way information is conveyed to the user on the screen, however, is a nonliteral manifestation of the software. The specific sequence of screens or the fact that by depressing the F10 key a help menu will appear are things that cannot be reduced to a tangible form. These user-interface elements are nonliteral manifestations of the program and are often referred to as part of the intangible "look and feel" of a program.

Most courts agree that the literal manifestations of computer software are copyrightable. Specifically, a program's source code, object code, and the flowcharts used to map out a program can all be protected by the Copyright Act.<sup>27</sup> However, courts have trouble

---

27. Case law is replete with decisions finding the source and object code protectable under the Copyright Act. See *Williams Elec. Inc. v. Artic Int'l, Inc.*, 685 F.2d 870, 876-77 (3d Cir. 1982) (object code copyrightable); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1243 (3d Cir. 1983) (source and object code copyrightable), *cert. dismissed*, 464 U.S. 1033 (1984); *Midway Mfg. Co. v. Strohon*, 564 F. Supp. 741, 750 (N.D. Ill. 1983) (source and object code copyrightable); *Digital Communications Assoc., Inc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 454 (N.D. Ga. 1987) (source and object code copyrightable); *Hubco Data Products, Corp. v. Management Assistance Inc.*, 219 U.S.P.Q. (BNA) 450, 454 (D. Idaho 1983) (object code copyrightable); *GCA Corp. v. Chance*, 217 U.S.P.Q. (BNA) 718, 720 (N.D. Cal. 1982) (source and object code copyrightable). The courts have also found that it is a violation to take a de-

dealing with copyrights for nonliteral manifestations of computer software.

The difficulty lies in the amorphous nature of "nonliteral" elements of computer programs. Unlike the written code of a program or a flowchart that can be printed on paper, nonliteral elements - including such elements as the overall organization of a program, the structure of a program's command system, and the presentation of information on the screen - may be less tangibly represented.<sup>28</sup>

Because of these difficulties, whether the nonliteral user-interface of computer software is protected under the Copyright Act has been left to judicial interpretation of the Act.

## II. THE PRESENT JUDICIAL INTERPRETATION OF SECTION 102 OF THE COPYRIGHT ACT AS IT IS APPLIED TO USER-INTERFACES

The question of whether the user-interface of computer software is copyrightable depends on how the court interprets the statutory scope of copyright protection as defined by section 102 of the Copyright Act.<sup>29</sup> Section 102 is divided into two paragraphs. Paragraph A provides specific language defining what can be copyrighted. Paragraph B provides a general caveat that must be followed when ascertaining whether something falls within the scope of copyrightability as defined in paragraph A. To determine whether the user-interface of computer software is copyrightable, under section 102 of Title 17 of the United States Code, the judiciary must interpret the individual language of section 102(a) in light of the general caveat of section 102(b).

### A. *Interpreting The Individual Language of Section 102(a) of the Copyright Act to Protect User-Interfaces*

The Copyright Act was drafted in 1976 and remains virtually unchanged except for two minor amendments added in 1980. The scope of copyright protection, however, was defined almost two centuries earlier. To understand the specific language of Section 102(a) requires a look back at the history of the scope of copyright protection.

---

tailed description of the solution to a problem, (which is what a flowchart is) and program it into computer source code. *Williams v. Arndt*, 626 F. Supp. 571, 578 (D. Mass. 1985); *Synercom Technology, Inc. v. Univ. Computing Co.*, 462 F. Supp. 1003, 1013 n.5 (N.D. Tex. 1978).

28. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 46 (D. Mass. 1990).

29. 17 U.S.C. § 102 (1988).

### 1. *The history of the statutory language protecting copyright*

Copyright protection was first offered in the Act of May 31, 1790.<sup>30</sup> In that Act, the scope of copyright protection was easily determined. Copyright protection was extended to "any map, chart, book or books already printed."<sup>31</sup> Specifically listing everything afforded copyright protection had its obvious advantages. However, whenever Congress decided that some other item belonged on the "copyrightable" list, it had to repeal or amend the Act. By 1909, after 119 years of repealing and amending,<sup>32</sup> Congress decided to generalize the scope of copyright protection by providing protection to "all the writings of an author."<sup>33</sup> This was a step in the right direction, except that Congress added a list of examples to help clarify what "writings of an author" were.<sup>34</sup> Scientific progress brought with it the need for more and more examples of what was copyrightable. Congress was still repealing and amending the current copyright legislation to append the list of examples.<sup>35</sup> After twenty years of hearings, study, debate, and redrafting, Congress came up with the 1976 Copyright Act and the language of section 102.<sup>36</sup>

Two 1980 amendments occurred after Congress created the National Commission on New Technological Uses of Copyrighted Works (CONTU)<sup>37</sup> to address potential issues not addressed in the bill that eventually became the 1976 Copyright Act.<sup>38</sup> Congress gave CONTU the following mandate:

---

30. Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124, 124 (repealed 1831).

31. *Id.*

32. In 1802, designs, prints, etchings and engravings were added to the list of copyright protected items. Act of Apr. 29, 1802, ch. 36, § 2, 2 Stat. 171, 171. This Act was repealed by the Act of Feb. 3, 1831, ch. 16, §§ 1, 14, 4 Stat. 436, 436, 439 which added "musical compositions" to the list. The 1831 Act was amended by the Act of Aug. 18, 1856, ch. 169, 11 Stat. 138, 139, which added "dramatic compositions" to the list. The 1856 Act was amended by the Act of Mar. 3, 1865, ch. 126, §§ 1, 2, 13 Stat. 540, 540 which added "photographs and the negatives thereof" to the list. The 1865 Act was repealed by the Act of July 8, 1870, ch. 180 § 86, 16 Stat. 198, 212 which added "statuary" and "models or designs intended to be perfected as works of the fine arts" to the list. This Act was repealed in 1909. *Lotus*, 740 F. Supp. at 47.

33. Act of Mar. 4, 1909, ch. 320, § 4, 35 Stat. 1075, 1076 (previously codified at 17 U.S.C. § 4, reprinted in 17 U.S.C.A. App. § 4 (West Supp. 1990); recodified 1947; repealed 1976).

34. *Id.* at § 5.

35. In 1912 Congress added "motion pictures" as a further example of "all the writings of an author." Act of Aug. 24, 1912, ch. 356, § 5(1)-(m), 37 Stat. 488, 488 (repealed 1976). In 1972 Congress added "sound recordings" to the list. Act of Oct. 15, 1971, Pub.L. 92-140, § 1(b), 85 Stat. 391, 391 (repealed 1976).

36. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp 37, 47 (D. Mass. 1990).

37. Act of Dec. 31, 1974, Pub. L. No. 96-517, 94 Stat. 3015 (1980).

38. *Lotus*, 740 F. Supp. at 49-50.



- (b) The purpose of the Commission is to study and compile data on:
  - (1) the reproduction and use of copyrighted works of authorship—
  - (A) in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information . . . .
- (c) The Commission shall make recommendations as to such purposes access to copyrighted works and to provide recognition of copyright owners.<sup>39</sup>

CONTU made its recommendations in a 1978 report.<sup>40</sup> CONTU determined that continued protection for computer programs was desirable,<sup>41</sup> but also that the 1976 Act already provided adequate protection with respect to the copyrightability of computer programs.<sup>42</sup> CONTU did not recommend any changes to the existing statutory language, but recommended two statutory amendments. The first defined what a computer program was<sup>43</sup> and the second allowed the owner of a computer program to make additional copies or adaptations of the program.<sup>44</sup> Congress readily adopted both recommendations.<sup>45</sup> The language from the 1976 Copyright Act, along with the 1980 amendments constitute the statutory law in the area of copyrights.

39. *Id.* (quoting the Act of December 31, 1974, Pub. L. 993-573, § 201(b)-(c), 88 Stat. 1873, 1873-74 (1974)).

40. NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT (1978) [hereinafter FINAL REPORT].

41. *Id.* at 20-21.

42. *Id.* at 16.

43. The National Commission on New Technological Uses of Copyrighted Works ("CONTU") recommended that "a 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." *Id.* at 14.

44. CONTU made the following recommendation:

Notwithstanding the provisions of section 106 [which grants the copyright owner the exclusive rights to reproduce the copyrighted work], it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

FINAL REPORT, *supra* note 40, at 31.

45. Congress amended section 101 of the 1976 Copyright Act to add CONTU's definition of a computer program verbatim. Congress also incorporated CONTU's second recommendation into section 117 of the Act. Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015, 3028 (codified at 17 U.S.C. §§ 101 and 117, respectively).

Like the 1909 Copyright Act, the 1976 Act included a general definition of what could be copyrighted and a list of clarifying examples.<sup>46</sup> The 1976 version however, defined what could be protected by the Act in more detail and included a list of examples in broad categories that could encompass several specific examples.<sup>47</sup>

## 2. *The specific language of section 102(a)*

The cornerstone provision of section 102 states that "copyright protection subsists . . . in original works of authorship fixed in any tangible means of expression . . ."<sup>48</sup> The scope of copyrightability can be restated as a work of authorship that meets the two fundamental criteria of originality and fixation in tangible form.<sup>49</sup> Interpreting the language of section 102(a) means interpreting the statutory meaning of "originality," "fixation in a tangible medium," and "work of authorship."

### a. The "original" language of section 102

The two tools that the judiciary uses when interpreting section 102 of the Copyright Act are the definitions found in section 101 and section 102's accompanying historical and statutory notes. Congress purposely left the term "original" undefined in section 101 of the Copyright Act,<sup>50</sup> but the House Report states that the standard used to judge an original work of authorship "does not include requirements of novelty, ingenuity, or aesthetic merit, and there is no intention to enlarge the standard of copyright protection to require

---

46. 17 U.S.C. § 102(a) (1976).

Copyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Works of authorship include the following categories: (1) literary works; (2) musical works, including any accompanying words; (3) dramatic works, including any accompanying music; (4) pantomimes and choreographic works; (5) pictorial, graphic, and sculptural works; (6) motion pictures and other audiovisual works; (7) sound recordings; and architectural works.

*Id.*

47. In the 1909 Act, copyright protection could secure "all the writings of an author." Act of Mar. 4, 1909, ch. 320, § 4, 35 Stat. 1075, 1076 (previously codified at 17 U.S.C. § 4, reprinted in 17 U.S.C.A. App. § 4 (West Supp. 1990); recodified in 1947; repealed 1976). The 1976 Act, by comparison, granted copyright protection for "original works of authorship fixed in any tangible medium of expression . . ." 17 U.S.C. § 102(a) (1988). The term "works of authorship" was much broader than the term "writings of an author." The effect was a broader scope of protection and less necessity for changing the Act.

48. 17 U.S.C. § 102(a) (1988).

49. H.R. REP. No. 1476 94th Cong., 2d Sess. 51, reprinted in 1976 U.S.C.C.A.N. 5659, 5664-65.

50. *Id.*

them.”<sup>51</sup> The court in *Puddu v. Buonamici Statuary Inc.* held that “originality” meant “only that the work owes its origin to the author.”<sup>52</sup>

b. The “fixed in a tangible means of expression” language of section 102

A “work of authorship” must be “fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.”<sup>53</sup> The term “fixed” is defined in Section 101 of the Act, but there is no definition for a “tangible medium of expression.” Section 101 states:

[A] work is “fixed” in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration. A work consisting of sounds, images, or both, that are being transmitted, is “fixed” for purposes of this title if a fixation of the work is being made simultaneously with its transmission.<sup>54</sup>

Congress intentionally drafted the statutory language concerning “any tangible medium of expression” broadly so that copyrightability will not be determined based on the form or medium in which a work is fixed.<sup>55</sup> The medium of fixation may be “in words, numbers, notes, sounds, pictures, or any other graphic or symbolic indicia, whether embodied in a physical object in written, printed, photographic, sculptural, punched, magnetic, or any other tangible form and whether it is capable of perception directly or by means of any machine or device now known or later developed.”<sup>56</sup> Copyrightability under section 102(a) depends not on the medium in which the work of authorship is fixed, but rather the work of authorship itself.

c. The “works of authorship” language of section 102

Since 1790 when only maps, charts, and books were protected by copyright, Congress has gradually expanded the scope of copyright protection. Different areas of subject matter have been effected by this expansion, but none so much as scientific discoveries and technological advances where there has been so much recent

---

51. *Id.*

52. *Puddu v. Buonamici Statuary, Inc.*, 450 F.2d 401, 402 (2d Cir. 1971) (quoting NIMMER, COPYRIGHT § 11 at 33 (1971)).

53. 17 U.S.C. § 102(a) (1988).

54. *Id.* at § 101(3).

55. H.R. REP. No. 1476, *supra* note 49, at 51, *reprinted* in 1976 U.S.C.C.A.N. at 5659, 5665.

56. *Id.*

growth. In the computer industry, for example, new advances generally allow for new methods of expression and a potential for new classifications as "works of authorship." Authors will continue to find new ways to express themselves, but "it is impossible to foresee the forms that these new expressive methods will take."<sup>57</sup> The language of section 102 "does not intend either to freeze the scope of copyrightable subject matter at the present stage of communications technology or to allow unlimited expansion into areas completely outside the present congressional intent."<sup>58</sup> Congress intended the term "works of authorship" to extend copyright protection to new methods of expression as they evolved.<sup>59</sup> Like the 1909 Copyright Act, section 102(a) includes a list of categories to help clarify the term "works of authorship."

Works of authorship include the following categories:

- (1) literary works;
- (2) musical works, including any accompanying words;
- (3) dramatic works, including any accompanying music;
- (4) pantomimes and choreographic works;
- (5) pictorial, graphic, and sculptural works;
- (6) motion pictures and other audiovisual works;
- (7) sound recordings; and
- (8) architectural works.<sup>60</sup>

Section 101 states that the term "including" is meant to be illustrative and not limitative,<sup>61</sup> meaning that the scope of "original works of authorship" is not exhausted by this list. The judiciary's present interpretation of section 102(a) provides copyright protection to the user-interface of a computer program.

### 3. *The judiciary's application of section 102 to nonliteral elements of a program*

Congress did not include "computer programs" as one of the categories in section 102; however, "computer programs," as defined by the Act, fall within the "literary works" category.<sup>62</sup> Literary works are defined in Section 101 as "works, other than audiovisual

---

57. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 48 (D. Mass. 1990) (quoting H.R. REP. NO. 1476, *supra* note 49, at 51, *reprinted in* 1976 U.S.C.C.A.N. at 5659, 5665).

58. H.R. REP. NO. 1476, *supra* note 49, at 51, *reprinted in* 1976 U.S.C.C.A.N. at 5659, 5664.

59. *Lotus*, 740 F. Supp. at 48.

60. 17 U.S.C. § 102(a)(1)-(8) (1988).

61. *Id.* at § 101(3).

62. *Lotus*, 740 F. Supp. at 49; *see also* H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. at 5659, 5667 ("The term 'literary works' . . . includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression for original ideas, as distinguished from the ideas themselves.")

works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied."<sup>63</sup> The literal elements of a computer program, the source and object codes, are works expressed in words or symbols and clearly fall within the category of literary works. The Act's definition of a computer program does not, however, cover the nonliteral elements of a program and courts struggle to determine to what extent a computer program's copyright extends to those nonliteral elements.<sup>64</sup>

The leading case addressing the issue of copyright protection for a work's nonliteral elements is *Whelan, Inc. v. Jaslow Dental Laboratory, Inc.*<sup>65</sup> The *Whelan* court found that the copyrights of literary works can be infringed even when there is no substantial similarity between the work's literal elements.<sup>66</sup> The *Whelan* court cited *Twentieth Century-Fox Film Corp. v. MCA, Inc.*,<sup>67</sup> which held that one can violate the copyright of a play or book by copying its plot or plot devices, as well as other cases to support the idea that copyright protection cannot be limited to the literal text of a work.<sup>68</sup> The court then found by analogy that the copyrights of computer programs can be infringed even absent copying of the literal elements of the program.<sup>69</sup> The court concluded that "copyright protection of computer programs may extend beyond the programs' code to their structure, sequence, and organization."<sup>70</sup> Since the *Whelan* decision, nearly every court that has ruled on the issue of the copyrightability of a program's sequence, structure, or organization has followed *Whelan*.<sup>71</sup> The *Whelan* court did not

---

63. 17 U.S.C. § 101(3) (1988).

64. *Digital Communications Assocs. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 454 (N.D. Ga. 1987).

65. 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 107 S. Ct. 877 (1987).

66. *Id.* at 1234.

67. 715 F.2d 1327, 1329 (9th Cir. 1983).

68. *Id.*

69. *Whelan*, 797 F.2d at 1234.

70. *Id.* at 1248.

71. See *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989) (nonliteral aspects such as "structure, sequence, and/or organization of the program" are copyrightable); *SAS Inst., Inc. v. S & H Computer Sys., Inc.*, 605 F. Supp. 816, 830 (M.D. Tenn. 1985) ("copying of the organization and structural details" can form bases for infringement); *Telemarketing Resources v. Symantec Corp.*, 12 U.S.P.Q.2d (BNA) 1991, 1993 (N.D. Cal. 1989) (copyright applies to overall structure and organization of a computer program); *Pearl Syst., Inc. v. Competition Elec., Inc.*, 8 U.S.P.Q.2d (BNA) 1520, 1524 (S.D. Fla. 1988) ("Copyright protection of computer software is not limited to the text of the source or object code."); *but see Plains Cotton Cooperative Ass'n v. Goodpasture Computer Serv., Inc.*, 807 F.2d 1256, 1262 (rejecting *Whelan's* protection for structure, sequence, and organization, court instead held that sequence and organization, where dictated by market forces, is a

however specifically extend a computer program's copyright protection to its screen displays.<sup>72</sup>

To give protection to screen displays and other user-interface elements, the courts generally use the "audiovisual works" category<sup>73</sup> of section 102(a).<sup>74</sup> "Audiovisual works" are works that "consist of a series of related images which are intrinsically intended to be shown by the use of . . . devices such as . . . electronic equipment . . . regardless of the nature of the material objects . . . in which the works are embodied."<sup>75</sup> A court first employed audiovisual copyright protection for elements of a user-interface to protect highly artistic computer-generated screens of video games.<sup>76</sup> Today, the "audiovisual works" category of section 102(a) is used to protect user-interface of commercial software with little or no artistic originality.<sup>77</sup> Audiovisual copyright protects "the appearances, executable images, and input formats produced by the software on the monitor screens and the sequence of keystrokes used to manipulate information or desired functions by the user."<sup>78</sup>

The "literary works" category<sup>79</sup> of section 102 is also used to provide protection for user-interface aspects of a computer program. The term "literary works" . . . includes . . . compilations of data. A compilation is a "work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work constitutes an original work or authorship."<sup>80</sup> After determining that a screen display would need its own copyright, the court in *Digital Communication Associates, Inc. v. Softklone Distributing Corp.* concluded that the status screen of a computer program was copyrightable as a compilation.<sup>81</sup> A "status screen" which is just one image displaying a message, menu, or other information, would be a "literary work"<sup>82</sup> if written on a piece of paper as opposed to the computer

---

non-copyrightable idea rather than a copyrightable expression) *reh'g denied*, 813 F.2d 407 (5th Cir. 1987), *cert. denied*, 484 U.S. 821 (1987).

72. *Digital Communications Assocs. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 455 (N.D. Ga. 1987).

73. 17 U.S.C. § 102(a)(6) (1988).

74. See *Telemarketing Resources v. Symantec Corp.*, 12 U.S.P.Q. 2d (BNA) 1991, 1995 (N.D. Cal. 1989); *Broderbund Software, Inc. v. Unison World Inc.*, 648 F. Supp. 1127, 1133 (N.D. Cal. 1986).

75. 17 U.S.C. § 101 (1988).

76. *Williams Elecs. v. Artic Int'l*, 685 F.2d 870 (3d Cir. 1982); *Kramer Mfg. Co., Inc. v. Andrews*, 783 F.2d 421 (4th Cir. 1986).

77. *Brown*, *supra* note 1, at 988.

78. *Id.*

79. 17 U.S.C. § 102(a)(1) (1988).

80. *Id.* at § 101.

81. *Digital*, 659 F. Supp. at 462 (N.D. Ga. 1987).

82. See 17 U.S.C. § 102(a)(1) (1988).

screen.<sup>83</sup> The *Digital* court concluded that a status screen was a "compilation" and within the scope of copyrightability as a section 102 "literary work."<sup>84</sup>

Section 102(a)(5) grants copyright protection to "pictorial, graphic, and sculptural works."<sup>85</sup> The section 101 definition of a "pictorial, graphic, and sculptural work" states in pertinent part:

[T]he design of a useful article . . . shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article.<sup>86</sup>

A "useful article" is an article "having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information."<sup>87</sup> In the words of the court in *Broderbund Software Inc. v. Unison World, Inc.*,<sup>88</sup> "copyright protection extends only to the artistic aspects, and not the mechanical or utilitarian features of a protected work."<sup>89</sup> The *Broderbund* court found that the structure sequence and layout of the audiovisual displays in question were dictated primarily by artistic and aesthetic considerations, and not by utilitarian or mechanical ones.<sup>90</sup> Thus, the court found that the user-interface aspects of a computer program were within the scope of "pictorial, graphic, and sculptural works," and, therefore, within the scope of section 102(a). The court in *Lotus Development Corp. v. Paperback Software International*<sup>91</sup> also found that audiovisual displays of a computer program were not merely utilitarian in nature but had expressive elements that could fall within the scope of "pictorial, graphic, and sculptural works" as well. "The point is that those elements of a useful article that can exist independently of the utilitarian aspects of the article are potentially copyrightable because those elements are elements of expression that can be distinguished from the utilitarian functions of the article."<sup>92</sup> The defendants in *Lotus* advanced a semantical argument stating that the screen displays were "articles," that they were 'useful' and that useful articles were not protected by copyright law.<sup>93</sup> The court found that the utilitarian aspects of "useful

---

83. *Digital*, 659 F. Supp. at 462.

84. *Id.* at 463.

85. 17 U.S.C. § 102(a)(5) (1988).

86. *Id.* at § 101.

87. *Id.*

88. 648 F. Supp. 1127, 1133 (N.D. Cal. 1986).

89. *Id.* at 1133.

90. *Id.*

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

92. *Id.* at 52.

93. *Id.* at 56-7.

articles" were not copyrightable, however it also stated that "it is not true . . . that every aspect of a user-interface that is 'useful' is therefore not copyrightable."<sup>94</sup>

Courts have found that the user-interface elements of computer software fall within the scope of section 102(a) as "audiovisual works," as "compilations" that are "literary works," and as expressive aspects of "useful articles" that fall under the "pictorial, graphic, and sculptural works" category of section 102(a).

### B. The General Caveat of Section 102(b)

The judiciary has interpreted the specific language of Section 102(a), the subject matter of copyright protection, to include the nonliteral structure, sequence, and organization of the program, and the user-interface elements of the program. However, as the court stated in *Mastro Plastic Corp. v. NLRB*,<sup>95</sup> the judiciary must "look to the provisions of the whole law" when interpreting statutory meaning.<sup>96</sup> When interpreting copyrightability under section 102, the court must look at the specific language of 102(a) in light of the general caveat of 102(b). Even if certain elements of computer programs are considered "works of authorship" under section 102(a), they may not be "entitled to an unlimited scope of copyright protection."<sup>97</sup> Section 102(b) states:

in no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it described, explained, illustrated, or embodied in such work.<sup>98</sup>

Computer programs are copyrightable only to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.<sup>99</sup> This distinction is often referred to as the "idea/expression dichotomy."

#### 1. The idea/expression dichotomy

The idea/expression dichotomy has its roots in the case of *Baker v. Selden*.<sup>100</sup> The *Baker* court held that the text of a book describing a special method of double-entry on paper spreadsheets (the T-accounts system) was copyrightable expression, but that the

---

94. *Id.* at 57.

95. 350 U.S. 270 (1956).

96. *Id.* at 285 (quoting *United States v. Heirs of Boisdore*, 49 U.S. 113, 122 (1850)).

97. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 49 (Mass. 1990).

98. 17 U.S.C. § 102(b) (1988).

99. H.R. REP. NO. 1476, *supra* note 49, at 54, 57, reprinted in U.S.C.C.A.N. at 5667, 5670. See also, *Lotus*, 740 F. Supp. at 53.

100. 101 U.S. 99 (1879).



method itself, which embodied the idea of this particular kind of double-entry bookkeeping, was not. The court concluded that Baker did not infringe Selden's copyright when Baker wrote his own treatise, in his own words, describing the special double-entry method of bookkeeping.<sup>101</sup> To encourage the flow of ideas, the judiciary protects the marketing of expression of a person's idea by enforcing a copyright on that expression. This provides a financial incentive for people to share their ideas. The ideas themselves, however, are unprotected. For example, two authors may have the same idea of canoeing through the Columbia River Gorge and writing a book about their experience. Even though their books are substantially similar, if one has not copied the expression of the other, there is no infringement. The second author may even have received the idea to write about canoeing through the Columbia River Gorge by reading the first author's book, but if the second author has not copied the first author's book, there is no copyright infringement.

An exception to the idea/expression dichotomy is the doctrine of merger. Some ideas can only be expressed in a limited number of ways. When this occurs, the idea and the expression are said to merge and the expression of the idea is not given copyright protection.<sup>102</sup> The leading case on the doctrine of merger is *Herbert Rosenthal Jewelry Corp. v. Kalpakian*.<sup>103</sup> In *Herbert*, the issue was whether or not the defendant infringed the plaintiff's copyright of a jeweled bee pin.<sup>104</sup> The court found that copyright protection was limited to the expression of the idea and not the idea itself, and that the plaintiff was free to take the idea of making jeweled pins in the shape of a bee. The court, however, also found that there was only one expression for the idea of a jeweled bee pin, namely, a jeweled bee pin.<sup>105</sup> The court noted that the idea and its expression appeared to be indistinguishable and ruled that the similarity between the plaintiff's and defendant's pins was inevitable.<sup>106</sup> The court ultimately held that where the idea and expression are inseparable, copying the expression must be permitted.<sup>107</sup>

The doctrine of merger applies not only when there is one ex-

---

101. *Id.*

102. See e.g., *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971); *Johnson Control Sys., Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989); *Digital Communications Assocs., Inc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 457-58 (N.D. Ga. 1987); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 59 (D. Mass. 1990); *Telemarketing Resources v. Symantec Corp.*, 12 U.S.P.Q. 2d (BNA) 1991, 1993 (N.D. Cal. 1989).

103. 446 F.2d 738 (9th Cir. 1971).

104. *Id.* at 739.

105. *Id.* at 742.

106. *Id.*

107. *Rosenthal*, 446 F.2d at 742.

pression for an idea as in the *Herbert* case,<sup>108</sup> but also when there are a limited number of expressions for a given idea. If a particular expression "is one of a quite limited number of the possible ways of expressing the idea . . . the expression is not copyrightable."<sup>109</sup> When the idea beyond the expression is such that it can be expressed in only a limited number of ways, then a party or number of parties could copyright all the ways to express the idea and effectively deny any future use of the idea.<sup>110</sup> Affording copyright protection to expressions of ideas where there is one or a limited number of expressions for the particular idea would grant a monopoly on the idea to the copyright owner.<sup>111</sup>

## 2. *The application of the idea/expression dichotomy to the nonliteral elements of a program*

In 1986, the court in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*<sup>112</sup> applied the idea/expression dichotomy to something other than a literal manifestation of a work for the first time.<sup>113</sup> The *Whelan* court drew the line between idea and expression at a place that provided copyright protection to the nonliteral structure and sequence of a program. The *Whelan* court, expounding upon *Baker v. Selden*, found that one way to distinguish idea from expression is to recognize that "the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea."<sup>114</sup> In *Bull HN Information Systems, Inc. v. American Express Bank Ltd.*,<sup>115</sup> the court, adopting the *Whelan* test, found that "to the extent structural similarities in a computer program are not necessary to the purpose or function of the program, they constitute a protectable expression of an idea."<sup>116</sup>

The court in *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*,<sup>117</sup> used a different test to distinguish between idea and expres-

---

108. See also, *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 59 (D. Mass. 1989); *NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d (BNA) 1177, 1188 (N.D. Cal. 1989).

109. *Lotus*, 740 F. Supp. at 59; cf., *Frybarger v. IBM Corp.*, 812 F.2d 525, 530 (9th Cir. 1987); *NEC*, 10 U.S.P.Q.2d at 1188.

110. *Lotus*, 740 F. Supp. at 59.

111. *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971).

112. 797 F.2d 1222 (3d Cir. 1986).

113. See *supra* notes 62-94 and accompanying text for a discussion of the application of § 102 to nonliteral elements of a program.

114. *Whelan, Inc. v. Jaslow Dental Lab. Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986).

115. No. 88 Civ. 2103 (SWK), 1990 WL 48098 (S.D.N.Y. 1990).

116. *Bull HN Infor. Sys., Inc. v. American Express Bank Ltd.*, No. 88 Civ. 2103 (SWK), 1990 WL 48098, at 3 (S.D.N.Y. 1990).

117. 886 F.2d 1173 (9th Cir. 1989).

sion. The *Johnson* court found that if some discretion or opportunity for creativity existed in the creation of the programs structure, the structure was copyrightable.<sup>118</sup>

The functionality test of the *Whelan* court and the creativity test of the *Johnson* court were combined by the court in *Digital Communications Associates, Inc. v. Softklone Distributing Corporation*<sup>119</sup> to distinguish copyrightable expression from noncopyrightable idea in the area of user-interfaces. The *Digital* court used the *Whelan* approach to find that "the use of a screen to reflect the status of the program is an 'idea,' the use of a command driven program is an 'idea,' and the typing of two symbols to activate a specific command is an 'idea,'" and hence not copyrightable.<sup>120</sup>

The legislative purpose of section 102(b) of the Copyright Act is to restate and codify that the basic dichotomy between expression and idea remains unchanged; expression is copyrightable, idea is not.<sup>121</sup> The one major exception to this rule is when a particular idea has a limited number of expressions. To grant a copyright to an expression under these circumstances would be to grant a copyright to the idea behind the expression.

Interpreting the language of section 102(a), the judiciary found that certain nonliteral manifestations of computer software, including the user-interface, are original works of authorship fixed in tangible means of expression. They have also found that these same nonliteral manifestations can be copyrightable expression, as opposed to noncopyrightable idea. Why is it then, that a debate exists over how the user-interface should be protected? Why do some courts hold that a program's copyright extends to protect the user-interface while others hold that the user-interface should have its own copyright?<sup>122</sup> This author believes that confusion exists because the judiciary has ignored the interdependent nature of computer software. The judiciary determined whether the structure of a program or the user-interface of a program was copyrightable by

---

118. *Johnson Control Sys., Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989).

119. 659 F. Supp. 449 (N.D. Ga. 1987).

120. *Digital*, 659 F. Supp. at 459.

121. H.R. REP. NO. 1476, *supra* note 49, at 57, *reprinted in* 1976 U.S.C.C.A.N. at 5659, 5670.

122. Some courts suggest that the program's overall copyright should protect the user-interface. See *Broderbund Software, Inc. v. Unison World*, 648 F. Supp. 1127 (N.D. Cal. 1986). Other courts suggest that the user-interface of a program should have its own copyright. *Digital Communications Assoc., Inc. v. Softklone Distrib. Corp.*, 659 F. Supp. 499 (N.D. Ga. 1987). In fact the court in *Manufacturers Technologies, Inc. v. CAMS, Inc.*, 706 F. Supp. 984 (D. Conn. 1989) created a legal fiction to suggest that there is a middle ground between these two theories.

looking only at those elements individually, and not by looking at the program as a whole. Had the judiciary taken a step back and recognized that computer programs are composite works with interdependent parts, their analysis would have been different.

### III. THE INTERDEPENDENT NATURE OF COMPUTER PROGRAMS AND THE COPYRIGHTABILITY OF USER-INTERFACES

The judiciary has questionably interpreted the language of section 102 to provide copyright protection to the nonliteral user-interface of computer software. Section 102 was not drafted with nonliteral expression in mind. Even the statutory definition of a computer program, by only referencing "statements and instructions,"<sup>123</sup> suggests that the statute recognizes that a program is made up of only literal manifestations. Nonetheless, the *Whelan* court extended copyright protection past the literal manifestation of a work and granted copyright protection to a nonliteral manifestation of a computer program for the first time. The *Whelan* court did not, however, address how or whether copyright protection should extend to the user-interface of a program.<sup>124</sup> Consequently, several theories emerged on how this nonliteral manifestation should be protected. Most of the theories are a combination of two main theories.<sup>125</sup> The first theory, called the extension theory, espouses the notion that the program's copyright, or the copyright of the literal manifestation of the program, should cover the nonliteral manifestations as well. The second theory referred to as the separate copyright theory, concludes that the user-interface of a program should have its own copyright. Under both the extension theory and the separate copyright theory, the interdependent nature of computer software prohibits the judiciary from finding the user-interface to be copyrightable under section 102.

#### A. *The Interdependent Nature of Computer Software Prevents the Judiciary From Finding That the Copyright of the Underlying Program Extends to Protect the Program's User-Interface*

The extension theory was applied in the case of *Broderbund Software, Inc. v. Unison World Inc.*<sup>126</sup> In *Broderbund*, Broderbund Software Inc. developed and copyrighted a program called Print Shop that created greeting cards, signs, posters, and banners.<sup>127</sup>

---

123. 17 U.S.C. § 101 (1988).

124. *Digital Communications Assoc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 455 (N.D. Ga. 1987).

125. See, e.g., Dorny & Friedland, *supra* note 5.

126. 648 F. Supp. 1127 (N.D. Cal. 1986).

127. *Id.* at 1130.

The Broderbund software was developed for use on Apple computers.<sup>128</sup> The defendant, seeing a market for a similar program for IBM compatible machines, developed a program called Printmaster.<sup>129</sup> The defendant's program had similar screen displays which led Broderbund to file for copyright infringement.<sup>130</sup> Because the court in *Broderbund* "discussed the screens and the underlying program as a unit, the court's opinion has generally been construed as holding that a copyright of a program also protects the screen displays generated by it."<sup>131</sup> The *Broderbund* court based its holding on the *Whelan* decision. The *Broderbund* court applied the reasoning of *Whelan*, which found that a program's structure was copyrightable, to find that the arrangement of screens was copyrightable also.<sup>132</sup>

### 1. *Whelan revisited*

The *Whelan* court dealt with an alleged infringement of the nonliteral structure sequence and organization of a computer program, but its holding is employed to find user-interfaces to be copyrightable expression. The *Whelan* court found that the "purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea."<sup>133</sup> The *Whelan* court's finding was based on its reading of *Baker v. Selden*.<sup>134</sup> The *Whelan* court stated that "as *Baker v. Selden*<sup>135</sup> focused on the end sought to be achieved by Selden's book, the line between idea and expression may be drawn with reference to the end sought to be achieved by the work in question."<sup>136</sup> Determining the desired end or purpose of a wholly literal work like that in *Baker v. Selden*, however, is very different from determining the purpose of a work with nonliteral elements.

In *Baker v. Selden*, Selden obtained a copyright on his book which described a new simplified system of accounting.<sup>137</sup> The book included blank forms for use in Selden's accounting system.<sup>138</sup>

---

128. *Id.*

129. *Id.* at 1130-31.

130. *Broderbund*, 659 F. Supp. at 1131.

131. Jeffrey R. Benson, Note, *Copyright Protection for Computer Screen Displays*, 72 MINN. L. REV. 1123, 1128 (1988).

132. *Broderbund Software, Inc. v. Unison World Inc.*, 648 F. Supp. 1127, 1133 (N.D. Cal. 1986).

133. *Whelan, Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986).

134. 101 U.S. 99 (1880)

135. *Id.*

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

137. *Baker v. Selden*, 101 U.S. 99, 102 (1880).

138. *Id.*

Baker was charged with infringing Selden's copyright by making and selling accounting books that used substantially the same system of accounting and a reproduction of the blank forms that Selden had included in the back of his book.<sup>139</sup> Both parties agreed that Selden's new method of accounting was uncopyrightable as a method or idea.<sup>140</sup> Both parties also agreed that the text portion of Baker's book did not infringe on Selden's copyright.<sup>141</sup> The issue was whether the blank forms that Baker reproduced and included in his book were "necessary incidents" to the method of accounting, or simply part of the text of the book.<sup>142</sup> If the method of accounting could not be used without the use of the blank forms, then the forms were "necessary incidents" and were not copyrightable. If they were considered part of the text, then they were expression, and the forms would be protected by the book's copyright.<sup>143</sup> The *Baker* court found that the forms were indeed "necessary incidents" and hence were part of the idea behind the method of accounting and not protected by the book's copyright.<sup>144</sup>

In *Whelan*, the Jaslow Dental lab felt that "the business operations of Jaslow Lab could be made more efficient if they were computerized."<sup>145</sup> Jaslow Lab hired Whelan to write a program to meet the needs of Jaslow Lab.<sup>146</sup> Whelan wrote a program for use on a certain IBM machine.<sup>147</sup> Jaslow Lab realized that there was a market for the program in offices with smaller computers and wrote another program for use on a variety of smaller computers.<sup>148</sup> The court found that Jaslow Lab's program "although written in a different computer language from [Whelan's program] and although not a direct transliteration of [Whelan's program], was substantially similar to [Whelan's program] because its structure and overall organization were substantially similar."<sup>149</sup> The *Whelan* court reached its decision by analogizing Ms. Whelan's program for the efficient management of a dental lab's finances, to Selden's method of accounting.<sup>150</sup> By finding that the purpose or desired end of the Whelan program was simply the management of a dental lab's finances, the *Whelan* court easily found that the structure and organ-

---

139. *Id.*

140. *Id.*

141. *Baker*, 101 U.S. at 102.

142. *Id.*

143. *Id.* at 103.

144. *Id.* at 104.

145. *Whelan, Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1225 (3d Cir. 1986).

146. *Id.*

147. *Id.*

148. *Id.* at 1226.

149. *Whelan*, 797 F.2d at 1228-29.

150. *Id.* at 1238.

ization of the program was not a "necessary incident"<sup>151</sup> and therefore copyrightable expression.<sup>152</sup> The purpose of Whelan's program, however, was *not* simply the management of a dental lab's finances. The *Whelan* court failed to consider the interdependent nature of computer programs and in failing to do so, drew an improper analogy.

## 2. *The purpose of a program*

The purpose of most commercial computer programs is comprised of three interdependent purposes corresponding to the three different parts of commercial software; the basic form, the structure and organization, and the user-interface.<sup>153</sup> The purpose of the "basic form" of a program is simply to be able to instruct the computer to perform tasks outlined in the flowchart. The purpose of the structure and organization of the program is different from, but dependent upon, the purpose of the "basic form" of the program. The purpose of the structure and organization of a program is efficiency. Efficiency is vital if the software is to be commercially successful. The purpose of the user-interface is different from both the purpose of the "basic form" of the program and the structure and organization of the program, but likewise, dependent on both of them. The purpose of a program's user-interface is to make the efficient "basic form" of the program easy for the average user to run. This usually entails writing additional code, often in a completely different programming language<sup>154</sup> than the "basic form" of the program.

The purpose or desired end of the Whelan program was not simply the management of a dental lab's finances, but rather an efficiently fast, easy to use program to manage a dental lab's finances. The *Whelan* court analogized the desired end or purpose of Selden's book to only part of the desired end or purpose of Whelan's program. A proper analogy to the desired end of the whole program reveals that the structure and organization and the user-interface are "necessary incidents" to the purpose of the program, and are therefore part of the idea of the program which is not copyrightable.

The *Broderbund* court held that the user-interface of a program was not necessary to the function of the program because the court found that another program had the same idea as

---

151. *Baker v. Selden*, 101 U.S. 99, 102 (1880).

152. *Whelan*, 797 F.2d at 1238.

153. *See supra* pp. 745-48 for a discussion of software's three interdependent parts.

154. User-interfaces are usually written in a "shell language" like "C-shell" which was designed specifically for the purpose of programming user interfaces.

Printmaster, but the other program had a different user-interface. The court concluded that this meant the particular user-interface of Printmaster was not inseparable from the program's idea, and was therefore, copyrightable. The court, however, made the same mistake that the *Whelan* court made. The court mistook the function of the overall program for the function of the "basic form" of the program. The court in essence found that the user-interface expression was separate from the idea behind the "basic form" of the program, and therefore, copyrightable. The court did not take into account the fact that the idea behind the overall program was a combination of the separate ideas behind the "basic form" of the program, the structure and organization of the program, and the user-interface of the program. The extension theory is applied by looking at the function or purpose of the "basic form" of the program to define what constitutes idea in the screens.<sup>155</sup> Because the idea behind the nonliteral user-interface is different from the idea behind the literal code of the "basic form" of the program, the screen displays will always be found to be separate from the idea behind the "basic form" and therefore always be found to be copyrightable. If a programmer writes a program that has a similar user-interface to another program, but which is generated by a different code in a different language, infringement may be found although none exists. The extension theory of protecting user-interfaces offers few guidelines to future programmers who must continue to be wary of infringing another's copyright even though they have written completely different code. Infringement will always be based on circumstantial evidence and such an application of Section 102 can only lead to endless litigation.<sup>156</sup>

*B. The Interdependent Nature of Computer Software Prohibits  
the Judiciary From Finding the User-Interface  
Copyrightable Under the Separate Copyright  
Theory*

A logical alternative to the extension theory is the separate copyright theory. If protecting user-interfaces with the copyright of the underlying program is troublesome, why not give the user-interface its own copyright? The *Digital* court used the separate copyright theory to provide copyright protection for user-interfaces. The court determined that computer screens should have their own copyrights after the manner of video games whose screens are sepa-

---

155. *Broderbund Software, Inc. v. Unison World Inc.*, 648 F. Supp. 1127, 1133 (N.D. Cal. 1986).

156. Peter G. Spivack, *Does Form Follow Functions? The Idea/Expression Dichotomy in Copyright Protection of Computer Software*, 35 UCLA L. REV. 723, 735 (1988).



rately copyrighted. The *Digital* court reasoned that a computer screen can be copied by a program that in no way copied the underlying program that generated the copied screen display.<sup>157</sup> Since the screen display was not a "direct copy" of the literary or substantive content of the computer program, the *Digital* court concluded that "a computer program's copyright protection does not extend to the program's screen displays and that copying of a program's screen displays, without evidence of copying of the program's source code, object code, sequence, organization or structure, does not state a claim of infringement."<sup>158</sup>

To determine whether user-interfaces are copyrightable under the separate copyright theory, courts generally apply the creativity test. The court in *Johnson Control Systems, Inc. v. Phoenix Control Systems, Inc.*<sup>159</sup> was one of the first courts to use the "creativity test" to find that a nonliteral part of a computer program was copyrightable expression. The *Johnson* court found that where some discretion or opportunity for creativity existed in the creation of the program's structure, the structure was copyrightable.<sup>160</sup> Applying the creativity test to user-interfaces, the court in *Lotus v. Development Corp. v. Paperback Software International and Stephenson Software, Ltd.*<sup>161</sup> found that because certain aspects of its user-interface were not used in every spreadsheet program, those aspects were the product of creativity and therefore copyrightable.<sup>162</sup> When determining whether the user-interface of a program is copyrightable expression, however, the courts must keep in mind the interdependent nature of a computer program.

The purpose or function of the user-interface is to make the basic form of the program easy to use. It is a "necessary incident"<sup>163</sup> to the overall program. The user-interface involves some creativity, but it is first and foremost functional in nature. The functional nature of the user-interface limits the creativity of the user-interface. To use an oft cited analogy, the user-interface is much like the "figure-H" pattern of a stickshift.<sup>164</sup> The pattern's sole purpose is to make the operation of shifting gears convenient or easy. The pattern is chosen with the fact that the car needs four

---

157. *Digital Communications Assoc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 455 (N.D. Ga. 1987).

158. *Id.* at 456.

159. 886 F.2d 1173 (9th Cir. 1989).

160. *Id.* at 1176.

161. 740 F. Supp. 37 (D. Mass. 1990).

162. *Id.* at 68.

163. *See supra* notes 142-152 and accompanying text for a discussion of "necessary incident."

164. *Synercom Technology, Inc. v. University Computing Co.*, 462 F. Supp. 1003, 1013 (N.D. Tex. 1978); cf. *Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv.*, 807 F.2d 1256, 1262 (5th Cir. 1987).

gear positions and a neutral position in mind. The pattern has an element of creativity in that the manufacture could have chosen a different figure that would have worked equally well. The "figure-K" pattern, for example, has positions that could correspond to the four gear positions and the neutral position. Other patterns such as the "+" symbol, or the "figure-x", or the "figure-I" could have worked as well. A pattern without five distinctive positions could also have worked, but would not have been as convenient, and the purpose for creating the gear shift pattern would have been foiled. It is clear that although the pattern selection involves some creativity, that creativity is limited by the functionality of the pattern. The H-pattern is necessary to the overall purpose of the gearshift assembly. Likewise, the user-interface is necessary to the overall purpose of the program. In *Baker v. Selden*,<sup>165</sup> the court found that the blank forms contained in the back of Selden's book were necessary incidents to Selden's specific method of accounting. As such, they were considered part of the method itself or the idea behind the method, and not part of the copyrightable expression contained in the book.<sup>166</sup> If the court had solely based its decision on whether there was the opportunity for discretion or creativity in designing the forms, they would have reached the wrong conclusion, because creating the forms involved a measure of creativity. The judiciary ignores the interdependent nature of the software and looks only at the creativity of the user-interface to incorrectly find that it is copyrightable expression.

The interdependent nature of computer software also inhibits the copyrightability of the user-interface under the separate copyright theory because of the doctrine of merger.<sup>167</sup> User-interfaces are different from literal forms of expression. The computer screen is limited in size, restricting the expression of ideas that appear on the screen. As a result, "there are only three basic types of interface styles: menus, command languages, and interactive design."<sup>168</sup> User-interfaces can also be limited by market factors. Computer screens often emulate standard forms in a particular industry.<sup>169</sup> Additionally, "considerations of efficiency, ease of use, and skills of

---

165. 101 U.S. 99 (1880).

166. *Whelan, Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986).

167. See *supra* notes 101-111 and accompanying text for discussion of the doctrine of merger.

168. Dornay & Friedland, *supra* note 5, at 205. "A 'Menu' is a list of items on the screen from which the user may choose one. 'Command languages' accept input from the user in a specific format which the user must know. 'Interactive Design' prompts the user for the necessary information based on prior entries. Many programs use some combination of styles." *Id.* at 205, n. 69.

169. See *Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv.*, 807 F.2d 1256, 1262 (5th Cir. 1987).

the intended audience can greatly restrict the viable interface choices."<sup>170</sup> Finally, the underlying program always limits the user-interface. The features of the underlying basic form of the program determine how that program is to be packaged by the user-interface. The user-interface is a means to an end. The end is user-friendliness for the underlying program. The end always dictates the means used to achieve that end. These limiting factors will cause the idea behind the user-interface of a program and the expression of the program to merge.

Protecting the user-interface in these instances denies the use of the underlying idea. This is not the purpose of copyright protection, and so when an idea and its expression merge, the expression is not granted copyright protection.<sup>171</sup> More likely, however, these limiting factors will lead courts to find infringement where none exists.

Adding to the already confused climate surrounding user-interfaces, the copyright office decided, after the *Digital* decision, to enforce its long-standing policy not to register separate copyrights for screen displays.<sup>172</sup> The copyright office permits a single registration of a computer program to extend copyright protection to the screen displays so long as they contain original creative authorship.<sup>173</sup> The interdependent nature of the software prohibits copyright protection for the user-interface under section 102 of the Copyright Act. It prohibits protection under the extension theory because the user-interface is a necessary incident to the idea of the overall program. Thus, the user-interface is inseparable from the idea and cannot be protected under section 102(b). The interdependent nature of software also prohibits protection under the separate copyright theory even if the copyright office were to allow separate registration, because the creativity or discretion involved in the user-interface is inherently limited. This leads to a merger of the idea behind the user-interface and its expression, making it unprotectable under section 102(b).

The scope of copyrightability under section 102 was drafted with literal manifestations in mind. Composite works like software, that have both literal and nonliteral manifestations that are dependent on each other, have given the judiciary trouble when the judiciary tries to apply copyright law to them. Decisions in this area

---

170. Dorny & Friland, *supra* note 5, at 205.

171. Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971).

172. The *Digital* decision came down at a time when the copyright office was granting separate copyright registrations to screen displays despite its own policy.

173. See generally, Ralph Oman, 36 PAT. TRADEMARK & COPYRIGHT J. (BNA) 155 (1988).

have become "increasingly contorted as they attempt to manipulate an existing statutory framework that is unsuited to the field. . . . At some point . . . it will be impossible to bend the existing legal framework any further."<sup>174</sup> The interdependent nature of software makes user-interfaces uncopyrightable under the language of section 102. In addition, due to the interdependent nature of user-interfaces, protecting them conflicts with the constitutional purpose of promoting the progress of science.

#### IV. THE INTERDEPENDENT NATURE OF SOFTWARE MAKES THE PRESENT JUDICIAL INTERPRETATION OF SECTION 102 AT ODDS WITH THE CONSTITUTIONAL PURPOSE OF THE COPYRIGHT ACT

When expounding statutory mandates, the judiciary must not limit its examination to the specific language of a provision or to the provisions of a statute as a whole, but must also look to the object and policy behind the statutory provisions.<sup>175</sup> The judiciary must determine whether the interpretation they have given a particular provision is consistent with the provision's intended purpose.

The purpose of granting copyright protection to works of authorship is not to reward the author, but "to serve the public welfare by encouraging authors . . . to generate new ideas and disclose them to the public."<sup>176</sup> Rewarding the author with personal gain is merely the best way courts advance public welfare.<sup>177</sup> When construing section 102 of the Copyright Act, the judiciary "must be faithful to the statutory language and mindful of both the ultimate goal of copyright law—the advancement of public welfare—and Congress' chosen method of achieving this goal—private reward to the individual author."<sup>178</sup> Courts must not draw the line between the copyrightable and the uncopyrightable elements of a computer program, or in other words, the line between a program's idea and its expression, in a way that discourages authors or programmers from generating new ideas and disclosing them to the public. Simply stated, "[a] copyright should not grant anyone more economic power than is necessary to achieve the incentive to create."<sup>179</sup>

Protecting the user-interface of a program does not provide the incentive to create. It discourages rather than promotes the

---

174. Dorny & Friedland, *supra* note 5, at 208.

175. *Mastro Plastics Corp. v. NLRB*, 350 U.S. 270, 285 (1956) (quoting *United States v. Heirs of Boisdore*, 49 U.S. 118, 122 (1850)).

176. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 52 (D. Mass. 1990).

177. *Mazer v. Stein*, 347 U.S. 201, 219 (1954).

178. *Lotus*, 740 F. Supp. at 53.

179. NATIONAL COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS 12 (1978).

"[p]rogress of Science and useful arts"<sup>180</sup> in the area of computer software by prohibiting innovators from improving or building upon the work of their predecessors. This may stifle innovation in the user-interface, and in the basic form of the program that generates the user-interface.

The fact that the "progress of science" benefits from allowing scientists and inventors to build on the work of their predecessors was recognized long ago.<sup>181</sup> Sir Isaac Newton declared, "If I have seen further it is by standing on [the] shoulders of Giants."<sup>182</sup> The court in *Lotus Development Corp. v. Paperback Software International and Stephenson Software, Ltd.*,<sup>183</sup> referring to Sir Isaac Newton's declaration, stated that "the legally relevant shoulders of programming giants are their ideas—and do not extend to all of their expressions."<sup>184</sup>

In the case of composite works such as software, where the user-interface is necessary to accomplish the desired task of the overall program, copyrighting the user-interface denies successors access to both the ideas and the expressions of the underlying program. Because programs are interdependent in nature, protecting one part of a program affects the other parts. If the effect of trying to protect one part of the program results in stifling innovation in another, the judiciary should think twice before granting copyright protection to the first part. The user-interfaces of one program may become an industry standard, thus, they can be used as shields against competitive programs. This prevents the ideas for those programs from being shared with the public. This clearly inhibits progress of science which is the constitutional purpose of the Copyright Act. The judiciary should not try to protect part of the program at the expense of another, especially if the part they are trying to protect exists only to serve a more important part.

User-interfaces make software easy for the lay person, or one who has no idea as to how or what the software is actually doing. The general user does not care how or what the software does. The general user merely wants to feel certain that if they push a certain key, they will get a certain result. The easier it is to get the result the better. Users will gravitate to the software that has the most user-friendly interfaces. Once a user-interface in a particular type of software is established as the most user-friendly, a "de facto industry standard" is created and programmers try to emulate that

---

180. U.S. CONST. art I, § 8, cl. 8.

181. *Id.*

182. ROBERT K. MERTON, ON THE SHOULDERS OF GIANTS: A SHANDEAN POSTSCRIPT 31 (1965) (quoting a letter from Sir Isaac Newton to Robert Hooke on February 5, 1675).

183. 740 F. Supp. 37 (D. Mass. 1990).

184. *Id.* at 78.

standard.<sup>185</sup> This in turn reinforces the particular user-interface as the industry standard. New programs will not be successful if their user-interfaces are below the accepted standard.<sup>186</sup> Once enough users have become accustomed to a certain user-interface when working with a particular type of software, they become reluctant to use even a better type of the same software if it means learning how it works. There is also a tremendous cost involved in retraining users. Additionally, "the first developer of successful software who is able to set an industry standard may now use the copyright law as a shield against competitive market forces."<sup>187</sup>

User-interfaces have a certain amount of creativity, but their primary purpose in the context of the overall program is functional. User-interfaces exist only to make the "basic form" of the program easy to use. Without the "basic form" of the program, there is no need for the user-interface. If this functional aspect becomes standardized and is allowed to be copyrighted, programmers will have to write completely different user-interfaces which users will not use. The innovative and creative ideas of those other programs will be lost. In this regard, computer programs are much like presents. The gift is packaged in wrapping paper that makes it more presentable to the receiver. If a certain kind of wrapping paper becomes the standard, and is the only kind of wrapping paper that the receiver will open, gifts in other wrapping paper will never be opened and the gifts will be lost. Programs, like presents are interdependent in nature. Programs consist of an underlying efficient "basic form" analogous to the gift, and packaged in a user-interface, analogous to the wrapping paper. Gifts are wrapped in a variety of ways, but the wrapping is limited by the size and shape and general features of the gift. Likewise, the user-interface is limited by the features of the program. Standardized user-interfaces, if copyright protected, deny programmers the opportunity to share innovative programming ideas with society. The question is whether we want innovation in wrapping paper, or innovation in the underlying gifts.

When interpreting a statutory provision, the judiciary should not lose sight of the purpose of the provision. The purpose of the Copyright Law is to encourage innovation. Computer software is a composite work, thus protecting the user-interface affects the rest of the software. One effect is that successors will be unable to improve the work of their predecessors without prohibitive costs. If the user-interface has become a *de facto* industry standard, then protecting the user-interfaces will be at the expense of the underly-

---

185. Anthony R. Barkume, *Proprietary Protection of Computer User Interfaces*, 64 ST. JOHN'S L. REV. 559, 578 (1990).

186. *Id.*

187. Brown, *supra* note 1, at 977.

ing program which generates the user-interface. Both of these results are against the constitutional purpose of the Copyright Act.

## VI. CONCLUSION

The user-interface of computer software is a very important part of an overall computer program. It is the link between the user and the internal workings of the program. It allows the ordinary computer user to access an intricate program and have that program instruct the computer to perform various tasks. It is important to the marketing of the overall program because users often judge a program by how easy the program is to use, not by the innovative way a program operates.

The user-interface elements of computer software fall within the scope of section 102(a).<sup>188</sup> When interpreting copyrightability under Section 102, the court must look at the specific language of 102(a) in light of the general caveat of 102(b). Section 102(b) states: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."<sup>189</sup> Computer programs are copyrightable only to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.<sup>190</sup>

Generally courts use two methods to distinguish idea from expression in the user-interface of a program. First, courts call the purpose or function of a program the program's idea. Everything that is not necessary to the purpose or function of the program is the program's expression. Second courts determine whether there is any discretion or creativity in choosing the user-interface. If there is, the user-interface is separate from the program's idea and thus, is copyrightable.

An exception to the idea/expression dichotomy is the doctrine of merger. Some ideas can only be expressed in a limited number of ways. When this occurs, the idea and the expression are said to merge and the expression of the idea is not given copyright protection.<sup>191</sup>

---

188. See *supra* notes 48-94 and accompanying text for discussion of section 102(a).

189. 17 U.S.C. § 102(b) (1988).

190. H.R. REP. NO. 1476, *supra* note 49, at 54, 57 reprinted in 1976 U.S.C.A.N. 5667, 5670. See also *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 53 (D. Mass. 1990).

191. *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971); *Johnson Control Sys., Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989); *Digital Communications Assoc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 457-58 (N.D. Ga. 1987); *Telemarketing Resources v. Sy-*

There are generally two schools of thought as to how a user-interface should be protected. The first is that the user-interface should be protected by the copyright of the overall program. The second is that the user-interface should have its own copyright. Had the judiciary acknowledged the interdependent nature of computer software, the judiciary would have found that neither school of thought could be applied to protect user-interfaces. Under the first theory, the user-interface is a necessary incident to the purpose of the overall program and under *Baker v. Selden*,<sup>192</sup> therefore, it is part of the program's idea. Ideas cannot be copyrighted under Section 102(b). Under the second theory, the user-interface's functionality limits its creativity so that the idea and expression merge. Under the merger doctrine, limited expressions are not granted copyright protection. Thus, neither school of thought justifies copyrighting user-interfaces.

The interdependent nature of computer software also makes protecting the user-interface at odds with the constitutional purpose of the Copyright Act. The purpose of copyright law is to promote innovation and progress. Interpreting Section 102 of the Copyright Act to protect the user-interfaces of computer programs denies innovators the opportunity to improve or build on the work of their predecessors. Thus, the resulting industry standards stifle productive competition.

Copyright protection "was not originally intended to prevent competitors from offering similar products."<sup>193</sup> Copyrightable expression should encourage innovation, not discourage it. User-interface designers themselves oppose copyright protection for user-interfaces because they believe such protection would be harmful, rather than helpful, to the industry.<sup>194</sup> Interpreting section 102 as providing copyright protection for user-interfaces denies programmers access to the work of their predecessors and limits progress in the area of this "useful science." Thus, copyrights should not protect user-interfaces.

---

mantec Corp. 1989 WL 200350 (N.D. Cal. 1989); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 59 (D. Mass. 1990).

192. 101 U.S. 99, 102 (1880).

193. Brown, *supra* note 1, at 987.

194. Pamela Samuelson & Robert J. Glushko, *Comparing the Views of Lawyers and User Interface Designers*, 30 JURIMETRICS 137 (1990). When Lotus Development Corporation, the developer of the software whose screens set the standard for spreadsheet software user-interfaces, brought a "look and feel" suit against a competitor, many programmers picketed Lotus' headquarters. The programmers feared that protection of the "look and feel" would make it harder to bring new products to market. Ironically, Lotus 1-2-3 may have been an improvement on an earlier spreadsheet program called Visicalc. Debra Schwartz & John Rosenberg, *Computing the Cost of Copyright: Programmers Fight "Look and Feel" Lawsuits*, NEWSWEEK, Aug. 27, 1990, at 52.



