Loyola of Los Angeles Law Review

Volume 24
Number 4 *Growth Management and the Environment in the 1990s*

Article 12

6-1-1991

Lotus Leaves Software Copyright in a Dream State: Defining Protection of the User Interface Following Lotus Development Corp. v. Paperback Software International

Craig A. Laidig

LOTUS LEAVES SOFTWARE COPYRIGHT IN A DREAM STATE: DEFINING PROTECTION OF THE USER INTERFACE FOLLOWING *LOTUS DEVELOPMENT CORP. V. PAPERBACK SOFTWARE INTERNATIONAL*

I. INTRODUCTION

Under copyright infringement analysis, computer programs are composed of two elements: (1) literal elements, which are the actual commands spelled out by the computer programmer;[1] and, (2) non-literal elements, which are the displayed result of the computer executing the literal commands.[2] Traditionally, courts have afforded copyright protection to all literal elements of programs, but only to certain non-literal elements such as the structure, sequence and organization of the program.[3] Because non-literal elements are only apparent when a program is operating, such elements pose difficult copyright issues for courts.

In *Lotus Development Corp. v. Paperback Software International*,[4] the District Court of Massachusetts held that copyright protection in the plaintiff's popular "Lotus 1-2-3" spreadsheet program extended to elements of the "user interface."[5] *Lotus* has been a closely watched decision with strong implications for the future of copyright development.[6]

Software has intrinsic utilitarian aspects and functional requirements not found in most other forms of expression protected by copyright.[7] If the breadth of copyright protection is extended too far, it will be difficult for software developers to improve programs created by other developers, and easy for software developers to create monopolies.[8] On

1. *See infra* notes 11-14 and accompanying text.

2. *See infra* notes 15-18 and accompanying text.

3. *E.g.*, Whelan Assocs. v. Jaslow Dental Laboratory, 797 F.2d 1222, 1224-25 (3d Cir. 1986) (held copyright protection of computer program extended beyond literal elements of program code and included non-literal elements of structure, sequence and organization of program), *cert. denied*, 479 U.S. 1031 (1987).

4. 740 F. Supp. 37 (D. Mass. 1990).

5. *Id.* at 68. The court adopted the plaintiff's description of the "user interface," which included such elements as the menus, including their structure and organization, the "long prompts" that appear at the bottom of the menu further describing the menu choices, "the screens on which they appear, the function key assignments, [and] the macro commands" and macro language found within the plaintiff's program. *Id.* at 63. See *infra* notes 69, 83 for further definition and explanation of the user interface and macro commands.

6. *See infra* notes 223-24 and accompanying text.

7. *See infra* notes 48-53 and accompanying text.

8. *Lotus*, 740 F. Supp. at 52-53.

the other hand, if the breadth of protection is too narrow, investments in the development of new programs will not be protected, thereby stifling advancements.[9]

This Note examines the *Lotus* court's analysis of prior copyright protection afforded computer programs and its application to this case. The author offers a modification of the analysis using existing case authority that would have more clearly defined the scope of copyright protection for program developers. Finally, the author urges that Congress enact the findings of the National Commission of New Technological Uses of Copyrighted Works and convene a new Commission to review the adequacy of copyright protection for computer programs.

II. A SIMPLE OUTLINE OF COMPUTER OPERATION

Even the most casual observer recognizes the essential role computers play in today's society: from transportation—controlling the engines of today's automobiles—to leisure activities such as home electronics. Computers[10] also help perform individual and work-related tasks by using commercial software programs on personal computers.

This Note focuses on source code programming, the literal requests that a programmer[11] makes using a program language.[12] Literal elements are exactly that—the actual commands spelled out, character-by-character, line-by-line, for the execution of a problem.[13] Copying of the literal elements of code is easily identified.[14] *Non*-literal elements, however, are

---

9. *Id.*

10. A "computer" as used in this Note is the combination of (1) microprocessors that compute information; (2) the necessary storage media; and (3) the equipment, such as a keyboard and display monitor, necessary to allow use by individuals. W. BUCKSBAUM, PERSONAL COMPUTERS HANDBOOK 69-70 (2d ed. 1984). This machinery, without more, is of no use to individual computer users; software (programs, procedures, rules and documentation) is needed to respond to individual requests and compute a response. J. ROSENBERG, DICTIONARY OF COMPUTERS, DATA PROCESSING, AND TELECOMMUNICATIONS 488 (1984).

11. A programmer is an individual that has knowledge of one or more "programming languages." J. ROSENBERG, *supra* note 10, at 407. The programmer takes the user's end-objective and designs, writes and tests a program in a computer language that the computer will recognize to perform that task. *Id.*

12. A program language is the syntax the programmer uses to create instructions the computer will recognize. MCGRAW-HILL DICTIONARY OF SCIENTIFIC AND TECHNICAL TERMS 1175 (1974). Examples of such languages are Pascal, a language for programmers that emphasizes particular aspects of programming referred to as "structured programming," J. ROSENBERG, *supra* note 10, at 379 and, "BASIC," a mnemonic for Beginner's All-purpose Symbolic Instruction Code, a program with a simple set of commands primarily designed for numerical applications, W. BUCKSBAUM, *supra* note 10, at 41.

13. Apple Computer v. Franklin Computer Corp., 714 F.2d 1240, 1248 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

14. The mere presence of identical code, however, does not automatically indicate that a

more difficult to define, and copying of them is more difficult for the courts to identify. The non-literal elements of a program have been collectively termed the "look and feel" or "user interface" of the program.[15] A trained programmer can read source code commands and understand what the machine's non-literal response will be.[16] As the courts have come to deal with the copyright protection afforded a computer program, it has become necessary to distinguish the literal from the non-literal elements of the program.[17]

By definition, non-literal elements of a program are those that are not literally written in the program. Practically, what this means, is that aspects of a program, which are not apparent to the observer by just looking at the program code itself, may be protected by copyright even though they become apparent only when the program operates on a computer. When there is more than one way to reach the same end, and those ways are hidden from view, problems in copyright protection arise.[18] It is the non-literal elements of programs that have given the courts the greatest difficulty in defining the extent of copyright protection.[19] These problems and the courts' inability to solve them are most apparent in *Lotus*.

---

program was infringed. See *infra* notes 186-96 and accompanying text for a discussion of non-infringing identical code.

15. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 62-63 (D. Mass. 1990).

16. *See Apple Computer*, 714 F.2d at 1243. The processor chip in the computer cannot read this language, so the source code is converted through a compiling program to object code, the language from which a machine recognizes each instruction. W. BUCKSBAUM, *supra* note 10, at 48.

17. *See, e.g., Apple Computer*, 714 F.2d at 1248-49 (software copyright protection extends to object code—symbols recognized by computer—in addition to programmers' source code).

18. A math example may clarify this. Consider the following equations:

Equation #1: $10 - 5 = Y$
Equation #2: $30 \div 6 = Y$

The *literal* elements of these math problems are quite different: in equation #1, five is subtracted from ten; in equation #2, thirty is divided by six. If these operations were calculated by a computer program that only displayed the value for Y, both programs would appear identical to the observer—he or she would only see "5" on the monitor. This display, a *non-literal* aspect of "the program," is identical.

In less theoretical language, a screen that flashes a command requesting data entry may be achieved in some cases using a single program command that causes the computer to flash that message. A program may also be written to display the request for a half second, then blank for a half second, display, blank, repeating until requested data has been entered. Again, the literal code to accomplish this is different, but the non-literal element, *displaying a flashing message*, is identical.

19. *See, e.g., Lotus*, 740 F. Supp. at 43-62 (discussing difficulty in defining point of idea—expression distinction and application to computer software).

III. THE BACKGROUND OF COMPUTER COPYRIGHT PROTECTION

The protection of an author's original works from duplication by others is authorized by the United States Constitution, which recognizes the need to "promote the Progress of Science and useful Arts, by securing for limited Times . . . the exclusive Right to their respective Writings and Discoveries."[20] By protecting original ideas and expression from duplication and use by others, the continued development of innovative ideas is encouraged.[21] Computer program protection is provided to a limited degree under patent law for the protection of original ideas and concepts,[22] and to a greater degree by copyright law for original works of authorship.[23] The best security, however, for computer program authors is copyright protection, because the copyright statute specifically identifies computer programs as protectable.[24]

Copyright protection subsists in a work from its creation until fifty years after the author's death.[25] This protection, however, is not unbridled; the copyright statute specifically does not grant protection 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."[26] Copyright protection therefore extends only to the creator's *expression* in all original works of authorship, including musical, dramatic, graphic, audiovisual and literary works.[27]

---

20. U.S. CONST. art. I, § 8, cl. 8.

21. Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975) ("The immediate effect of our copyright law is to secure a fair return for an 'author's' creative labor. But the ultimate aim is, by this incentive, to stimulate artistic creativity for the general public good.").

22. *See* 35 U.S.C. §§ 1-376 (1988). It is likely that most software does not meet the originality requirements for patent protection. Davidson, *Protecting Computer Software: A Comprehensive Analysis*, 23 JURIMETRICS J. 337, 357 (1983).

23. *See* 17 U.S.C. §§ 101-914 (1988).

24. *See id.* §§ 101, 117. Other considerations include greater expense and difficulty procuring patent protection and a longer period of protection under copyright. *See* Sumner, *The Copyright/Patent Interface: Patent Protection for the Structure of Program Code*, 30 JURIMETRICS 107 (1989) (providing thorough discussion of patent versus copyright protection for computer software).

25. 17 U.S.C. § 302(a).

26. *Id.* § 102(b). While section 101 of the Copyright Act of 1976 is dedicated to the definition of terms used in the Act, none of the specific exceptions under the Act are defined in that section. *See id.* § 101.

27. *Id.* § 102(a). Additionally, the works 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." *Id.*

A. Recognition of Difficulties with Computer Copyright Law

The types of authorship included for copyright protection have gradually expanded as computer technology has grown.[28] Congress first identified programs as "copyrightable" in 1964 when the Register of Copyrights[29] indicated that computer programs would be accepted for registration.[30]

Congress revised the copyright laws by enacting the Copyright Act of 1976 (the 1976 Copyright Act).[31] During the formulation of the new laws, Congress created the National Commission on New Technological Uses of Copyrighted Works (CONTU or the Commission).[32] Congress formed the Commission to allow adequate study of the problems raised by computer uses that had not been dealt with in the Senate or House of Representatives bills, thereby avoiding further delay of the general copyright revisions.[33]

1. CONTU's final report

The findings of the Commission were not included in the 1976 Copyright Act because the Commission was not ready to publish its final report (*CONTU Report*) until July 31, 1978.[34] The *CONTU Report* recommended that Congress amend section 101 of the 1976 Copyright Act to add a definition for computer programs,[35] and replace the existing sec-

---

28. NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 1, 3 (1978) [hereinafter CONTU REPORT].

29. The general administrative functions and duties regarding the Copyright Act are the responsibility of the Register of Copyrights. 17 U.S.C. § 701(a) (1988). The Register of Copyrights is appointed by the Librarian of Congress. *Id.*

30. CONTU REPORT, *supra* note 28, at 15 n.73 (citing REGISTER OF COPYRIGHTS, COPYRIGHT OFFICE CIRCULAR No. 31D (Jan. 1965)).

31. Pub. L. No. 94-553, 90 Stat. 2541 (codified as amended at 17 U.S.C. §§ 101-810 (1976)).

32. Act of Dec. 31, 1974, Pub. L. No. 93-573, §§ 201-208, 88 Stat. 1873, 1873. Congress explained its function:

> (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, and . . .
> (2) the creation of new works by the application or intervention of such automatic systems or machine reproduction.
> (c) The Commission shall make recommendations as to such changes in copyright law or procedures that may be necessary to assure for such purposes access to copyrighted works, and to provide recognition of the rights of copyright owners.

Id. § 201, 88 Stat. at 1873-74.

33. CONTU REPORT, *supra* note 28, at 3.

34. *Id.* at 1.

35. The report suggested that "section 101 be amended to add the following definition: A 'computer program' is a set of statements or instructions to be used directly or indirectly in a

tion 117 of the 1976 Copyright Act with a new provision to allow specific copying for archival and utilization purposes peculiar to computer operation.[36] When it again revised the copyright laws, in the Copyright Act of 1980 (the 1980 Copyright Act),[37] Congress specifically afforded protection to computer programs by adopting the aforementioned recommendations from the Commission's report.[38] The *CONTU Report* also recommended that: "Any legislation enacted as a result of these recommendations should be subject to a periodic review to determine its adequacy in the light of continuing technological change. This review should especially consider the impact of such legislation on competition and consumer prices in the computer and information industries . . . ."[39] No review has been conducted to date.

2. Limitations placed on protection: The idea/expression dichotomy and utilitarian objects

During the latter part of the nineteenth century, the United States Supreme Court, in *Baker v. Selden*,[40] recognized that copyright protection is limited to the author's expression only and does not cover the author's ideas.[41] This restriction has since been incorporated into the

---

computer in order to bring about a certain result." CONTU REPORT, *supra* note 28, at 12. This suggestion was accepted without change by Congress. *See* Act of Dec. 12, 1980, Pub. L. No. 96-517, § 10(a), 94 Stat. 3015, 3028 (codified as amended at 17 U.S.C. § 101 (1988)).

36. CONTU REPORT, *supra* note 28, at 12. Section 117 reads in part:

> [I]t 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.

17 U.S.C. § 117 (1988).

The *CONTU Report* used the term "rightful possessor of a copy," which Congress, in its only change, replaced with "owner." *Compare* CONTU REPORT, *supra* note 28, at 12 ("[I]t is not an infringement for the *rightful possessor* of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program . . . ." (emphasis added)) *with* 17 U.S.C. § 117 ("[I]t 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 . . . ." (emphasis added)).

37. Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015 (codified as amended at 17 U.S.C. §§ 101, 117 (1980)).

38. H.R. REP. NO. 1307, 96th Cong., 2d Sess., pt. 1, at 23-24, *reprinted in* 1980 U.S. CODE CONG. & ADMIN. NEWS 6460, 6482-83.

39. CONTU REPORT, *supra* note 28, at 2.

40. 101 U.S. 99 (1880).

41. *Id.* at 105 (seminal case holding blank forms reproduced in author's book on account-

copyright statute.[42] However, deciding at what point an idea stops and its expression begins is quite difficult,[43] and having codified the restriction has not made the task any easier for the courts.[44]

The problems courts face in distinguishing ideas from creative expression were explained by Judge Learned Hand in *Nichols v. Universal Pictures Corp.*[45] In what has become the standard for defining the separation of the two concepts, Judge Hand described a theatrical play as follows:

> Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his [or her] "ideas," to which, apart from their expression, his [or her] property is never extended. Nobody has ever been able to fix that boundary, and nobody ever can.[46]

This concept of abstraction, without any guideline explaining where on the spectrum the distinction lies, has been adopted by many courts, including the court in *Lotus Development Corp. v. Paperback Software International*.[47]

Courts have also grappled with the utilitarian aspects of computer programs.[48] Section 101 of the 1976 Copyright Act includes the definition of a "useful article" as "having an intrinsic utilitarian function that

---

ing did not carry copyright protection on idea of double entry accounting). In discussing the difference between the author's text and the reproduced forms accompanying it, the Court stated that "[t]he object of the [author's text] is explanation; the object of the [forms] is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters patent." *Id.* at 105; *accord* Mazer v. Stein, 347 U.S. 201, 217-18, *reh'g denied*, 347 U.S. 949 (1954).

42. *See* 17 U.S.C. § 102(b) (1988); *supra* note 26 and accompanying text.

43. *See, e.g.*, Atari, Inc. v. North Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 616-17 (7th Cir.) ("[T]hat a work is copyrighted says very little about the scope of its protection."), *cert. denied*, 459 U.S. 880 (1982).

44. *See* Apple Computer v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983) ("Many of the courts which have sought to draw the line between an idea and expression have found difficulty in articulating where it falls."), *cert. dismissed*, 464 U.S. 1033 (1984).

45. 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931).

46. *Id.* at 121 (citations omitted).

47. 740 F. Supp. 37 (D. Mass. 1990). "[T]he legal test for deciding copyrightability . . . has not been precisely defined either in the copyright statute or in precedents interpreting and applying it." *Id.* at 59.

48. *See, e.g., Apple Computer*, 714 F.2d 1240; Apple Computer v. Formula Int'l, 562 F. Supp. 775 (C.D. Cal. 1983), *aff'd*, 725 F.2d 521 (9th Cir. 1984); Data Cash Sys. v. JS & A

is not merely to portray the appearance of the article or to convey information. An article that is normally *a part* of a useful article is considered a 'useful article.' "[49] Useful articles are not considered protectable expression under copyright analysis.[50] While a computer program could be considered a non-protectable useful article by this statutory definition, the term "computer program" is also defined in the statute.[51] Section 117 of the 1980 Copyright Act indicates that computer programs are protected under copyright law.[52] The statutory inclusion of protection for computer programs, combined with the express exclusion of utilitarian objects, which programs can be, complicates the courts' analyses of how far copyright protection extends.[53] The difficulties courts face in dealing with the utilitarian aspect of computer programs has contributed to the inconsistencies in prior decisions, thereby laying the foundation for the ill-defined policy established in *Lotus*.

B. The Developing Case Law

The Commission recognized the difficulty in separating expression from other non-copyrightable elements of computer programs such as the process being performed by the program.[54] The *CONTU Report* stated that the line between copyrightable form and uncopyrightable process "should be drawn on a case-by-case basis by the institution designed to make fine distinctions—the federal judiciary."[55]

Little guidance is available to the courts to assist them in determin-

---

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

49. 17 U.S.C. § 101 (1988) (emphasis added).

50. Brandir Int'l v. Cascade Pacific Lumber Co., 834 F.2d 1142, 1146-47 (2d Cir. 1987).

51. "A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101.

52. Section 117 states that copies made for archival purposes or those necessitated in the utilization of the program are not infringements. *Id.* § 117.

53. *See, e.g.*, Whelan Assocs. v. Jaslow Dental Laboratory, 797 F.2d 1222, 1236 n.28 (3d Cir. 1986) ("This test [to separate expression from utilitarian idea] is necessarily difficult to state, and it may be difficult to understand in the abstract."), *cert. denied*, 479 U.S. 1031 (1987).

54. "It is difficult, either as a matter of legal interpretation or technological determination, to draw the line between the copyrightable element of style and expression in a computer program and the process which underlies it." CONTU REPORT, *supra* note 28, at 22. The report continued: "[N]ew applications which advancing technology will supply may make drawing the line of demarcation more and more difficult." *Id.*

55. *Id.* at 22-23. The Commission's findings were probably supported by prior case decisions on this problem. *See, e.g.*, Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931). The *Nichols* court stated that "[w]hile we are as aware as any one that the line, wherever it is drawn, will seem arbitrary, that is no excuse for not drawing it; it is a question such as courts must answer in nearly all cases." *Id.* at 122.

ing when a program *is* a copyrightable expression and when it is a non-protectable idea, procedure or process.[56] The meager congressional record available only reiterates, in even more general terms, the limits stated by CONTU.[57] Finding sparse legislative history for an essentially "bare-bones" statute, some courts considering copyright protection for complex computer programs have relied upon the *CONTU Report* to infer legislative intent.[58] Understandably, judicial review has meandered through intricate computer programs searching for factual elements with which a copyright infringement analysis can be fashioned.

In its report, the Commission found that "[f]low charts, source codes, and object codes are works of authorship in which copyright subsists."[59] Holding that a program is a literary work, the Court of Appeals for the Third Circuit, in *Apple Computer v. Franklin Computer Corp.*,[60] found that copying of the literal elements of a program, even if only to allow compatibility with other vendors' programs, was an infringement.[61] This holding was subsequently extended to include the non-literal elements of a program.[62]

The line between expression and non-protectable process that CONTU had discussed appears to have been drawn by a Fifth Circuit district court in *Synercom Technology v. University Computing*.[63] In *Synercom*, the order and sequence of data input formats were held to be

56. *See* Atari, Inc. v. North Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 615 (7th Cir.) ("[T]here is no litmus paper test by which to apply the idea-expression distinction; the [court's] determination is necessarily subjective."), *cert. denied*, 459 U.S. 880 (1982).

57. "[Section 102(b) indicates] that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law." H.R. REP. NO. 1476, 94th Cong., 2d Sess. 57, *reprinted in* 1976 U.S. CODE CONG. & ADMIN. NEWS 5659, 5670.

58. *See, e.g.*, Micro-Sparc, Inc. v. Amtype Corp., 592 F. Supp. 33 (D. Mass. 1984); Atari, Inc. v. JS & A Group, 597 F. Supp. 5 (N.D. Ill. 1983). *But see* Whelan Assocs. v. Jaslow Dental Laboratory, 797 F.2d 1222, 1241-42 (3d Cir. 1986) (court held *CONTU Report* "cannot be a substitute for legislative history. . . . with respect to provisions not amended [to the 1976 Copyright Act] in response to the Report"), *cert. denied*, 479 U.S. 1031 (1987).

59. CONTU REPORT, *supra* note 28, at 21. Flow charts are block diagrams showing the programmer the outline of the problem that the program must solve. *Id.* n.109. Source code is the program in a language which a programmer can observe and understand, directing the machine to execute the steps necessary for the commands identified in the flow chart. J. ROSENBERG, *supra* note 10, at 490. Object code is the machine language which instructs the processor to perform particular functions created from the source code. *Id.* at 355.

60. 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

61. *Id.* at 1253.

62. *See* SAS Inst. v. S & H Computer Sys., 605 F. Supp. 816, 826 (M.D. Tenn. 1985) (defendant "slavish[ly]" copied literal, non-literal and organizational elements of plaintiff's program into another language to run on different computer).

63. 462 F. Supp. 1003 (N.D. Tex. 1978).

non-protectable ideas, not expression.[64] This apparent dividing line, however, was soon erased by the Court of Appeals for the Third Circuit in *Whelan Associates v. Jaslow Dental Laboratory*.[65] In *Whelan*, a program written in a different computer language than the program for which the plaintiff sought protection, and with substantial differences in its literal code, was found similar to the plaintiff's program for copyright purposes.[66] The *Whelan* court held that copyright protection goes beyond literal elements of the program to the structure, sequence and organization of the program itself.[67] This holding has been adopted in several circuits.[68]

In January 1987, Lotus Development Corporation (Lotus) brought a suit against Paperback Software International (Paperback) and Stephenson Software, Limited for infringing the copyright on the user interface[69] of its "Lotus 1-2-3" (1-2-3) electronic spreadsheet program.

---

64. *Id.* at 1013. The court analogized the problem to the "Figure-H" shift pattern of a manual automobile transmission:

> The [shift] pattern (analogous to the computer "format") may be expressed in several different ways: by a prose description in a driver's manual, through a diagram, photograph, or driver training film, or otherwise. Each of these expressions may presumably be protected through copyright. But the copyright protects copying of the particular expressions of the pattern, and does not prohibit another manufacturer from marketing a car using the same [shift] pattern.

Id.

65. 797 F.2d 1222 (3d Cir. 1986).

66. *Id.* at 1229.

67. *Id.* at 1239.

68. *See, e.g.*, Digital Communications Assocs. v. Softklone Distrib. Corp., 659 F. Supp. 449 (N.D. Ga. 1987) (Eleventh Circuit); Dynamic Solutions, Inc. v. Planning & Control, Inc., 1987 Copyright L. Dec. (CCH) ¶ 26,062 (S.D.N.Y. Feb. 2, 1987) (Second Circuit); Broderbund Software v. Unison World, 648 F. Supp. 1127 (N.D. Cal. 1986) (Ninth Circuit); E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485 (D. Minn. 1985) (Eighth Circuit). *But see* Plains Cotton Coop. v. Goodpasture Computer Serv., 807 F.2d 1256 (5th Cir.) (expressly rejecting *Whelan* where sequence and organization may be dictated by marketplace), *cert. denied*, 484 U.S. 821 (1987).

69. The *Lotus* court described the term "user interface" of 1-2-3 as including such elements as the menus (including their structure and organization), the "long prompts" that appear at the bottom of the menu further describing each menu choice, the screens on which they appear, the function key assignments, and the macro commands and macro language. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 63 (D. Mass. 1990). The term "look and feel," offered before trial by the plaintiff, was apparently abandoned at the time of trial. *Id.* at 62-63.

IV. THE *LOTUS* COURT'S APPROACH TO THE BREADTH OF COMPUTER SOFTWARE COPYRIGHT

A. Facts of Lotus Development Corp. v. Paperback Software International

The concept of an electronic spreadsheet[70] was first put into commercial practice in a computer program called VisiCalc.[71] While VisiCalc proved to be a commercial success, its marketability as a spreadsheet was limited because it was written to operate only on the Apple II personal computer.[72] With the introduction of the IBM personal computer, the Apple system's limitations became more apparent.[73] "Building on [VisiCalc's] revolutionary idea for an electronic spreadsheet,"[74] the original authors of Lotus 1-2-3 took advantage of the IBM personal computer enhancements and created "an evolutionary product that was built upon the shoulders of VisiCalc."[75] The 1-2-3 program went on to become a great success; it is the leading spreadsheet program sold today.[76]

Dr. James Stephenson, the founder of one of the co-defendants, Stephenson Software, also recognized the limitations of VisiCalc.[77] Stephenson began development of his own electronic spreadsheet, eventually released as VP-Planner.[78] During the initial development stages, the menus used in this program were different from those used in both VisiCalc and 1-2-3.[79] One month before installing the initial version of VP-Planner, however, Stephenson observed the 1-2-3 program and continued to improve upon his own program for the next year and a half.[80]

By the autumn of 1984, the success of 1-2-3 was apparent. Stephen-

---

70. A spreadsheet "is an electronic replacement for the traditional financial modeling tools: the accountant's columnar pad, pencil, and calculator." D. COBB & G. LEBLOND, USING 1-2-3, at 17 (2d ed. 1986).

71. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 65 (D. Mass. 1990). Electronic spreadsheets have found tremendous use not just in the financial community, but in all areas where many computations have to be made, as they produce nearly instantaneous results. D. COBB & G. LEBLOND, *supra* note 70, at 9-12.

72. *Lotus*, 740 F. Supp. at 65.

73. *Id.* The Apple computer had limited memory, limited screen display and a limited number of keys on the keyboard (*e.g.*, no function keys and no up and down cursor arrows). When VisiCalc was rewritten for the IBM personal computer (IBM PC), it did not take advantage of these additional features of the IBM PC. *Id.*

74. *Id.* at 66.

75. *Id.* The original authors of Lotus 1-2-3 were Mitchell Kapor and Jonathan Sachs. *Id.*

76. Huber, *Madonna Ain't Software*, FORBES, Sept. 3, 1990, at 104.

77. *Lotus*, 740 F. Supp. at 68.

78. *Id.*

79. *Id.*

80. *Id.* at 68-69.

son decided that the success of VP-Planner depended on its compatibility with 1-2-3.[81] Later, in his affidavit to the district court, Stephenson stated: " 'The only way to accomplish this result . . . was to ensure that *the arrangement and names of commands and menus in VP-Planner conformed to that of Lotus 1-2-3*.' "[82] This would allow the transfer between spreadsheets without the loss of macros[83] or without requiring the retraining of their operators.[84]

The defendants then converted VP-Planner into a program more like 1-2-3, that was publicly advertised as a "workalike" for 1-2-3.[85] In fact, the operating manual for the program stated, "VP-Planner is designed to work like Lotus 1-2-3, keystroke for keystroke. . . . VP-Planner's worksheet is a feature-for-feature workalike for 1-2-3. It does macros. It has the same command tree. It allows the same kind of calculations, the same kind of numerical information. Everything 1-2-3 does, VP-Planner does."[86]

The *Lotus* court did, however, find some differences between VP-Planner and 1-2-3. For instance, the VP-Planner menu line begins with a different command than 1-2-3, and some menus include additional commands not found in 1-2-3.[87] Additionally, the start-up screens, location on the screen of menu lines, exact wording of the screen prompts, organization of "help" screens, width of the displayed screen and security of portions of the spreadsheet differ between the two programs.[88] Nevertheless, the court found the works "substantially, indeed, strikingly, similar."[89]

The court noted that Excel, another electronic spreadsheet program that is not exactly compatible with 1-2-3, has also been a commercial success.[90] Although Excel's main screen display more closely resembles a paper spreadsheet than does that in 1-2-3, the court found there to be "a rather low limit, as a factual matter," to the number of ways to make a computer screen resemble a spreadsheet.[91] The court extended this ar-

81. *Id.* at 69.
82. *Id.* (quoting Stephenson Affidavit) (emphasis added by court).
83. Macros are small programs within the spreadsheet usually created by the operator to execute a series of common computations or functions, for example, taking values from various locations in the spreadsheet and computing their sum. *Id.* at 72.
84. *Id.* at 69.
85. *Id.*
86. *Id.* at 69-70.
87. *Id.* at 70.
88. *Id.*
89. *Id.*
90. *Id.* at 69.
91. *Id.* at 66.

gument to find that the identification of "cells"[92] by means of a "rotated 'L' " configuration was therefore non-protectable idea, not expression.[93]

All parties in the *Lotus* dispute agreed, as a general proposition, that literal elements of a computer program, including both the source code and object code, if original, are copyrightable.[94] Lotus contended, however, that copyright protection extends to all elements of a program that consist of original expression, whether literal or non-literal, including expression embodied in a program's "user interface."[95]

B. The Lotus *Court's Legal Analysis—A New Test*

The *Lotus* court applied a new three-step test to determine whether it should afford copyright protection to the user interface.[96] Although the *Lotus* court discussed the background of facts and law in the computer program copyright area for eighteen pages,[97] it concluded that "the legal test for deciding copyrightability, in a factual context such as is presented here, has not been precisely defined either in the copyright statute or in precedents interpreting and applying it."[98] What exactly the court found so unique in the facts before it, however, is not clear.[99] The

92. A cell is the intersection of a column and row on the screen in which a numerical value, formula or text may be entered. *Id.* at 63.

93. *Id.* at 66. The court explained that the individual cells, the regions in which numbers or math formulas are entered, are generally identified by a "rotated 'L' " configuration (an "L" rotated 90 degrees clockwise). *Id.* at 63. Columns are identified across the top using the alphabet and rows down the left side using numbers. *Id.* Therefore, cell A1 would be the cell or box in the topmost left hand corner. In addition, a range of three cells down the left side would be the range from A1 to A3, and the range of three cells across the top would be the range from A1 to C1.

94. *Id.* at 45.

95. *Id.* at 45-46. See *supra* note 69 for a definition of the "user interface." The difference between the "user interface" and the rest of a program was well stated in Manufacturers Technologies v. CAMS, Inc., 706 F. Supp. 984 (D. Conn. 1989). The *Manufacturers Technologies* court explained: "The computer program and any authorship contained therein is designed to organize and direct the computer to efficiently perform a particular task when properly directed by the user," whereas "the user interface is designed to communicate with the user in a way to facilitate the understanding and use of the program itself." *Id.* at 993.

96. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 59-61 (D. Mass. 1990).

97. *Id.* at 45-62.

98. *Id.* at 59.

99. Elements of the user interface had previously been reviewed by the courts for copyright protection. *See supra* notes 62-68. The *Lotus* court first discussed Synercom Technology v. University Computing, 462 F. Supp. 1003 (N.D. Tex. 1978). *Synercom's* holding conflicted with the *Lotus* court's protection of the user interface. *Lotus*, 740 F. Supp. at 54-56. The *Lotus* court, with little support for its finding, dismissed *Synercom*, stating:

> *Synercom*, though, was published less than a month after the publication of the CONTU report (which it never cites) and well before the 1980 amendments. Since then, *congressional and judicial development* of the law of copyrightability of computer programs has advanced considerably, and *Synercom's* central proposition—

court merely stated that to gain copyright protection, a software developer must satisfy the court in each of the following three steps:

> FIRST, [the decision maker must] choose some formulation—some conception or definition of the "idea"—for the purpose of distinguishing between the idea and its expression. . . . SECOND, the decision maker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of *that* idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of that idea. . . . THIRD, having identified elements of expression not essential to every expression of the idea, the decision maker must focus on whether those elements are a substantial part of the allegedly copyrightable "work."[100]

The court suggested that its test was built on a solid foundation by asserting it to be within the guidelines of the many " 'markers' of the borderline between copyrightability and non-copyrightability."[101] Nevertheless, the court then directly contradicted long-followed case law when it declared that "idea" does not have to be separable from "expression" to afford copyright protection to the work.[102] Having watered down one of the few explicit findings in copyright idea/expression analy-

---

that the expression of non-literal sequence and order is inseparable from the idea and accordingly is not copyrightable—has been explicitly rejected by several courts.

Id. at 55 (citations omitted) (emphasis added). In fact, there is essentially no congressional development, *see supra* notes 56-58 and accompanying text, and the *Lotus* court's premise for this analysis is the lack of prior judicial definition, *Lotus*, 740 F. Supp. at 59. The reasoning for diminishing *Synercom's* holding for not referencing the *CONTU Report* is neither apparent nor explained. Following a lengthy discussion of holdings both for and against the protection of the user interface, the *Lotus* court made a conclusory dismissal of *Synercom's* holding. *See id.* at 55 ("In any event, *Synercom's* input formats are quite different from, and distinguishable from, the non-literal aspects of 1-2-3 at issue in this case.").

100. *Id.* at 60-61. The court recognized that the decision maker measures substantiality in this part of the test in both a qualitative and quantitative manner. *Id.* (*citing* SAS Inst. v. S & H Computer Sys., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985)).

101. *Id.* at 59.

102. *Id.* at 60. Although the court stated that "[o]ne need not totally disentangle the idea from its expression in order to conclude that a particular aspect is expression," *id.*, it has long been held that expression that is not separable from the idea may not be afforded copyright protection. Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971) (idea of jeweled bee pin inseparable from expression; protecting expression in such circumstances grants monopoly on idea to copyright owner); *see also* Baker v. Selden, 101 U.S. 99, 103 (1879) (methods and diagrams necessary incidents to expression are not protectable); Morrissey v. Procter & Gamble Co., 379 F.2d 675, 678-79 (1st Cir. 1967) (where number of ways to express are limited, no particular form of expression comes from the subject matter and copyright may not be granted). The *Lotus* court cited *Herbert Rosenthal* for this very premise. *Lotus*, 740 F. Supp. at 59.

ses, the court immediately backed down from its rigorous legal test in the first part of its three-step analysis. The *Lotus* court held that the decision maker must focus upon "alternatives that counsel may suggest, *or the court may conceive*,"[103] and the decision maker should then choose some formulation along the scale of abstraction discussed by Judge Hand in *Nichols v. Universal Pictures Corp.*[104] This fundamental step in deciding a copyright infringement case—identification of the idea from which protectable expression is derived—and the *first* step in the *Lotus* court's legal test, was never clearly made. Inferentially, however, the court identified the "electronic spreadsheet" as the idea from which the expression derived.[105]

---

103. *Lotus*, 740 F. Supp. at 60 (emphasis added).

104. 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931); *see supra* note 46 and accompanying text.

105. *See Lotus*, 740 F. Supp. at 65-66. The *Lotus* court seemed to start its analysis by identifying the most fundamental level of abstraction under Judge Hand's test. *See id.* at 65. The court discussed the non-protectable idea of an electronic spreadsheet. *Id.* Although not clear, the court apparently adopted this level of abstraction for the remainder of its analysis. The court stated:

> At *the most general level* of Hand's abstractions scale the computer programs at issue in this case, and other computer programs that have been considered during the course of trial, are expressions of the idea of a computer program for an *electronic spreadsheet*. . . . Thus, even though programs like VisiCalc, 1-2-3, Multiplan, SuperCalc4, and Excel are very different in their structure, appearance, and method of operation, each is, at the most basic level, just a different way of expressing the same idea: the electronic spreadsheet.

Id. at 65 (citations omitted) (emphasis added). The court further stated:

> Of course, if a particular expression of *the idea of an electronic spreadsheet* communicates no details beyond those essential to stating the idea itself, then that expression would not be copyrightable. The issue here is whether Lotus 1-2-3 does go beyond those details essential to any expression of the idea, and includes substantial elements of expression, distinctive and original, which are thus copyrightable.

Id. (emphasis added). With no further delineation of the "idea" being expressed other than this most general abstraction, the court began its analysis of the expressive elements, having apparently adopted this general abstraction as the conceptual "idea" under the first step of the legal test. The court continued:

> Accordingly, [the rotated "L" screen display] of *electronic spreadsheet computer programs*, if not present in every *expression of such a program*, is present in most expressions. . . . *Another expressive element that merges with the idea of an electronic spreadsheet*—that is, that is an essential detail present in most if not all expressions of an electronic spreadsheet—is the designation of a particular key that, when pressed, will invoke the menu command system.

Id. at 66 (emphasis added). The court at this point discussed the menu command system as "expression" of an electronic spreadsheet; however, the court had previously considered the menu command system to be "idea" and not "expression." The court observed:

> The *idea* for a two-line moving cursor menu is also functional and obvious, and, indeed, is used in a wide variety of computer programs including spreadsheet programs. Nevertheless, it does not follow that every possible method of designing a menu system that includes a two-line moving cursor is non-copyrightable.

Id. at 65 (emphasis added). The court's repeated reference to the idea of an electronic spread-

1. The court's flawed abstraction of protectable expression

The defendants in *Lotus* asserted that only literal manifestations of computer programs are copyrightable.[106] Lotus, on the other hand, asserted that copyright protection extends to all elements of a computer program containing original expression, whether literal or non-literal, including the expression found in a program's "look and feel" or user interface.[107] The *Lotus* court selectively reviewed the *CONTU Report* and the sparse congressional record, ultimately concluding that "Congress manifested an intention to use the idea-expression distinction as part of the test of copyrightability for computer programs."[108] The court expressly rejected the defendants' literal versus non-literal test.[109]

The literal versus non-literal distinction was not an issue *de novo* for copyright analysis. In *Whelan Associates v. Jaslow Dental Laboratory*,[110] the Court of Appeals for the Third Circuit held that copyright protection goes beyond the literal elements of a program to the program's structure, sequence and organization.[111] In *Whelan*, a programmer brought a copyright infringement action against a dental laboratory for which a custom program had been developed.[112] Jaslow Dental Laboratory's program allowed operation on a different computer system.[113] This custom program was essentially identical to the plaintiff's custom program, but was written in another computer language.[114] The *Whelan* court found that while the literal code was substantially different, there was overall similarity between the two programs, and therefore infringement had occurred.[115]

Finding support in *Whelan*—although previously stating this was a new issue for copyright[116]—the *Lotus* court quickly dispensed with holdings that disagreed with *Whelan*. Without reviewing the merits of the holding, the court reasoned that refusing to invoke copyright protection

---

sheet, however, imputes this to be the formulation, conception or definition of "idea" that was made under the first step of its legal test.

106. *Id.* at 45.

107. *Id.* at 45-46.

108. *Id.* at 54.

109. *Id.*

110. 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

111. *Id.* at 1248.

112. *Id.* at 1225.

113. *Id.* at 1226.

114. *Id.*

115. *Id.* at 1248; *see also* Johnson Controls, Inc. v. Phoenix Control Sys., 886 F.2d 1173 (9th Cir. 1989) (non-literal aspects of program that are copyrightable are those that embody expression rather than idea).

116. *See supra* note 98 and accompanying text.

for the non-literal elements of sequence and ordering of data, as the court did in *Synercom Technology v. University Computing*,[117] would, as a practical matter, result in computer programs not getting the substantial protection the *Lotus* court found Congress to have mandated.[118] Furthermore, the *Lotus* court cited, but did not discuss, the Fifth Circuit's decision in *Plains Cotton Cooperative v. Goodpasture Computer Service*.[119] The *Plains Cotton* court, rejecting the *Whelan* decision, held that where the market dictates a standard format for a computer program, that format is not copyrightable expression.[120] The *Lotus* court later dismissed the defendants' claims of a market basis for the restriction of copyright on the program.[121]

The *Lotus* court stated that expression merges with the idea of an electronic spreadsheet where an essential detail is present in most, if not all, expressions of an electronic spreadsheet.[122] Under this analysis, necessary expression will not be extended protection. For example, the court discussed the use of the slash key ("/") to invoke the menu command system.[123] Eliminating all the letter, number and arithmetic keys as choices,[124] the court found the slash key to be "one of very few practical options."[125] Because the choice of available keys is so limited, the *Lotus* court found this expression to have merged with the idea of having a readily available method of invoking the menu command system. As a result, the use of the slash key to invoke the menu command system was held not protectable.[126]

Still, the *Lotus* court concluded that the menu command structure itself, as an element of the program capable of being expressed in an al-

---

117. 462 F. Supp. 1003 (N.D. Tex. 1978).

118. *Lotus*, 740 F. Supp. at 56.

119. *Id.* at 55 (citing Plains Cotton Coop. v. Goodpasture Computer Serv., 807 F.2d 1256 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987)).

120. *Plains Cotton*, 807 F.2d at 1262.

121. *Lotus*, 740 F. Supp. at 75-79. *See infra* notes 148-52 and accompanying text.

122. *Lotus*, 740 F. Supp. at 66.

123. *Id.* The court also identified the math function keys (+,−,×,/) as expression essential to spreadsheets. *Id.*

124. These keys are required for the use of an "electronic spreadsheet" and cannot also be used to invoke the menu systems. *Id.*

125. *Id.*

126. *Id.* The court named only the semicolon and slash keys as practical alternatives. *Id.* The court's argument that a user should not have to press two keys at the same time, such as the "Alt," "Shift" or "Ctrl" keys, is weakened by the plethora of programs that utilized such a system at the time of the court's decision. *See, e.g.*, WORDPERFECT CORP., WORDPERFECT 4 (Version 5.1 1989); MICRORIM, INC., R:BASE USER'S MANUAL ix (Version 3.0 1990); MERIDIAN TECHNOLOGY, INC., CARBON COPY PLUS 3-18 (1988).

most unlimited number of ways, was eligible for protection.[127] The court found the use of the terms "Worksheet," "Range," "Copy," "Move," "File," "Graph," "Data" and "Quit" to be non-obvious, and a unique way of expressing a command structure.[128] Apparently, to the court, simply spelling out and explaining command terms similar to the commands used in the VisiCalc program is not obvious.[129] Nevertheless, it is difficult to envision a term other than 1-2-3's "graph" to identify as clearly a command to display a graph.[130] The court, however, did not clarify where along the continuum of expression-abstraction the 1-2-3 menu system became protectable expression.

Compounding the confusion, the *Lotus* court found that Paperback had copied the macro command language of 1-2-3.[131] This determination was made in spite of the following findings of the court:

(1) The macro language requires the recognition of "cells." The court recognized that there were limited numbers of ways of recognizing cells, and that the system used in 1-2-3, a rotated "L," *was not copyrightable expression*.[132]

(2) Where the number of keys available for an action is limited, it merges expression to the idea and is not protectable.[133] Menu choices, using a unique character for each word, are limited by their nature to the number of choices. As such, this *would not be protectable expression*.[134]

(3) Mathematical operators, such as the addition, subtraction, multiplication and division symbols *are not copyrightable*.[135]

(4) The natural laws of mathematics, which are not copy-

127. *Lotus*, 740 F. Supp. at 68. The court at this point had switched to identifying the "idea" to be expressed as a *menu command structure*. *See supra* note 105 and accompanying text.

128. *Lotus*, 740 F. Supp. at 68.

129. VisiCalc used a menu system that identified choices only by the first letter of the term. *Id.* at 67. In one instance, the term was identical—VisiCalc displayed "M" for "move" and 1-2-3 displayed "Move" for "move." *See id.* at 67.

130. Other terms could be available, for example, "show," but word limits are essentially the same as the limits placed on the particular key invoking the menu system—a limitation the *Lotus* court found too restrictive to be afforded copyright protection. *See supra* note 126.

131. *Lotus*, 740 F. Supp. at 78.

132. *Id.* at 66. See *supra* note 92 for an explanation of the "rotated 'L'" system.

133. *Lotus*, 740 F. Supp. at 66.

134. *Id.* (court found keys to invoke menu *system* limited to few practical choices and not copyrightable). Although the court held differently, there is suggestion in the holding that terms in a menu may not be copyrightable, as they are obvious or merge with the idea of the particular command. *Id.* at 67.

135. *Id.* at 66.

> rightable, *force certain orders* of implementing cells, operators and choices described above.

The court then found that the macro language of 1-2-3,[136] utilizing this analysis, was copyrightable.[137] Yet, there is nothing in the copyright statutes or in subsequent court holdings that finds copyrightable expression resulting from a forced arrangement of non-copyrightable elements.[138]

2. Erroneous protection of the user interface resulting from the flawed analysis

The *Lotus* court adopted the proposition that the "look and feel," or user interface, includes: (1) the menu structure and organization, (2) the long prompts,[139] (3) the screen on which they appear, (4) function key assignments, and (5) the macro commands and language.[140] As a result, the *Lotus* court granted protection to each such component of the user interface. In each case, however, the court was wrong. First, menu structure and organization should not be afforded blanket copyright protection since it may contain only non-protected ideas and necessary expression.[141] Second, long prompts—menu descriptions—and the screens on which they appear are necessary functions of the menus; they are ideas used throughout the computer programming industry[142] and, as such, should not be subject to copyright protection.[143] The macro commands and language similarly should not be protected by copyright because they are fixed orders of non-copyrightable elements.[144]

Finally, function key assignments should not be protected. As-

---

136. "In 1-2-3, the command terms within a macro may consist of menu choices (*e.g.*, "/C" to copy a cell . . .)[,] keyboard commands (such as function keys, cursor keys, or the 'enter' key), and special macro commands . . . ." *Id.* at 64-65. While the special macro commands might contain terms of unique expression, the court did not distinguish this, instead focussing on a discussion of macros utilizing menu choices discussed in the text. *See id.* at 65.

137. *Id.* at 68.

138. CONTU recognized this limitation and discussed the probable application of parallel cases using historical facts, where the use of necessary facts is a "fair use" of copyrighted material and not an infringement. CONTU REPORT, *supra* note 28, at 23 (Commission cited, *e.g.*, Rosemont Enters. v. Random House, Inc., 366 F.2d 303 (2d Cir. 1966), *cert. denied*, 385 U.S. 1009 (1967)).

The United States Supreme Court recently stated: "Common sense tells us that 100 uncopyrightable facts do not magically change their status when gathered together in one place." Feist Publications v. Rural Tel. Serv. Co., 59 U.S.L.W. 4251, 4252 (Mar. 27, 1991).

139. See *supra* note 69 for an explanation of "long prompts."

140. *Lotus*, 740 F. Supp. at 63.

141. *See supra* notes 122-26 and accompanying text.

142. *Lotus*, 740 F. Supp. at 65.

143. *See infra* notes 186-92 and accompanying text.

144. *See supra* notes 131-38 and accompanying text.

signing actions to function keys within a program is severely restricted because most personal computers are limited to either ten or twelve choices for function key assignments.[145] The *Lotus* court recognized that copyright protection is unavailable when there is a limited choice of alternatives.[146] Had the *Lotus* court analyzed separately whether or not each component of the user interface was protectable, it would have concluded, based on the reported facts and the analyses it recognized in its opinion, that no copyright protection was available.[147]

3. The court's dispensation of marketplace disruption claims

Finally, the *Lotus* court ignored Paperback's claim of a need to achieve compatibility and standardization.[148] Paperback argued that granting copyright protection to a user interface will frustrate the public interest in allowing programmers to achieve innovation by "borrowing" and improving upon ideas of other programmers.[149] The court discounted this argument in part by citing other programs that were successful despite a different user interface.[150] Additionally, the court found no support for this proposition in the legislative history of the copyright statutes.[151] The court stated that "the statute does not bar copyright-

145. WORDPERFECT CORP., *supra* note 126, at 332.

146. *Lotus*, 740 F. Supp. at 66.

147. This is based on the concept-idea-processes of 1-2-3 identified by the *Lotus* court. *See id.* at 49. Certainly, elements with these processes may be protectable expression such as the wording placed in a submenu explaining a choice or even organization of menu commands *not* based on alphabetical order or frequency of use.

148. *See id.* at 77.

149. *Id.*

150. *Id.* at 78. The court cited the success of Excel, "an innovative spreadsheet program that is not compatible with 1-2-3, either in its menu structure or in its macro command facility." *Id.* Unfortunately, the court neglected to mention the fact that Excel gained its success on a completely different operating system, Apple Macintosh computers, and was not a competitor with 1-2-3 until recently. *See* E. JONES, USING EXCEL FOR THE PC 16 (1988). While other electronic spreadsheet programs have survived in the market, their success pales when compared to 1-2-3's market share. M. CAMPBELL, 1-2-3 POWER USER'S GUIDE 130 (1988). Even Excel, the leading spreadsheet on the Macintosh system, E. JONES, *supra*, at 16, included capabilities to read and write 1-2-3 files in its IBM version "[b]ecause of 1-2-3's wide base of support in corporate America," M. CAMPBELL, *supra*, at 130.

In a similar disregard for the realities of software operation and the software marketplace, the court stated that Paperback could have incorporated a macro conversion capability "as the creators of Excel have successfully done." *Lotus*, 740 F. Supp. at 69. Excel does contain a conversion program, however, its usefulness can best be described as marginal. *See, e.g.*, C. TOWNSEND, MASTERING EXCEL ON THE IBM PC 509 (1988) (window properties, 1-2-3 charts to Excel graphs and macros do not convert to Excel); E. JONES, *supra*, at 491 ("Due to many differences between Lotus 1-2-3 and Excel, some macros cannot be successfully translated.").

151. *Lotus*, 740 F. Supp. at 77-79.

ability merely because the originality of the expression becomes associated, in the marketplace, with usefulness of the work to a degree and in dimensions not previously achieved by other products on the market."[152]

This absolute finding is not supported by either case law or the legislative background of computer copyright. In *Plains Cotton*, the Fifth Circuit held that sequence and organization of program menus may constitute unprotectable "ideas" where market factors play a significant role in their determination.[153] Furthermore, in the *CONTU Report*, the Committee was cognizant of the potential interference within the computer industry that copyright protection could precipitate.[154] The testimony forming the foundation of the *CONTU Report* was heard between 1975 and 1978 when the computer industry was very different than it is today.[155] Yet, even at that point in time, the Committee noted:

> One of the hallmarks of a competitive industry is the ease with which entrepreneurs may enter into competition with firms already doing business. The absence of significant barriers to entering the program-writing market is striking. There are several hundred independent firms whose stock in trade is computer programs. New software firms may be formed with few people and little money; entry into the market has thus far been fairly easy. None of the evidence received by the Commission suggests that affording copyright to programs would in any way permit program authors to monopolize the market for their products. *Nor is there any indication that any firm is even remotely close to dominating the programming industry.*[156]

The Commission made clear in the *CONTU Report* that it was concerned with the potential burden of copyright on computer programs, devoting an entire section to a discussion of the economic effects of program copyright.[157] The Commission concluded from prior copyright cases that: "A copyright owner may monopolize his [or her] expression but not the market in which it is purveyed."[158] Additionally, in its initial recommendations, the Commission stated the objectives of computer copyright protection:

> To provide reasonable protection for proprietors without unduly burdening users of programs and the general public, the

152. *Id.* at 58.
153. *Plains Cotton*, 807 F.2d at 1262.
154. CONTU REPORT, *supra* note 28, at 12.
155. *Id.* at 1.
156. *Id.* at 23 (footnotes omitted) (emphasis added).
157. *Id.* at 23-25.
158. *Id.* at 24.

following statements concerning program copyright ought to be true:

1. Copyright should proscribe the unauthorized copying of these works.
2. Copyright should in no way inhibit the rightful use of these works.
3. Copyright should not block the development and dissemination of these works.
4. Copyright should not grant anyone more economic power than is necessary to achieve the incentive to create.[159]

It seems clear that the *Lotus* court overlooked the very real concerns of the Commission on the breadth of copyright protection to be implemented.

V. A MORE REASONABLE ANALYSIS OF SOFTWARE COPYRIGHT PROTECTION UNDER EXISTING AUTHORITY

Blanket protection of the non-literal elements of computer programs does not serve the purpose of copyright law—to encourage the development of original expression.[160] Programmers should be encouraged to write programs that will operate faster, use less memory or otherwise be more efficient than existing programs. A review of prior holdings demonstrates that the facts of the *Lotus* case are not unique. A new legal test was not required. A careful review of case precedent suggests that: (1) programs are comprised of expression for *many* "ideas"; (2) elements of programs, both literal and non-literal, may be "stock and trade" for programmers and should not warrant copyright protection; (3) screen displays should be protected *if* they convey information or expression not necessary to the idea being conveyed; and, (4) factors in the marketplace, in limited instances, may be considered by the court.

A. Identifying the "Copyright Idea" in a Computer Program

When identifying the "idea" behind a program in order to analyze the protectable "expression" of that idea, courts should abstain from formulating the idea at the most general level of abstraction as the court did in *Lotus*.[161] The most abstract level of idea is overly broad and includes

159. *Id.* at 12.

160. *See* Harper & Row Publishers v. Nation Enters., 471 U.S. 539, 546 (1985); Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975).

161. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990); *see supra* note 105.

isolated ideas within that abstraction, erroneously granting copyright protection to ideas.[162] Part of the policy of copyright protection—that of the free use of ideas[163]—assumes that the similarity required for a finding of infringement varies according to the type of work and the ideas being expressed.[164] The *Lotus* court did not recognize this, thereby utilizing a grossly overbroad analysis.

In *Nichols v. Universal Pictures Corp.*,[165] Judge Learned Hand's abstraction test decreased the detail of two plays being compared until a point was reached where the two works shared common elements.[166] Judge Hand found no infringement by the defendant's play where the common matter was only found at the most general, single idea of the trouble and turmoil of an inter-religious marriage.[167] Even though Judge Hand found only this single idea common to the works, other ideas were considered by the court, including the incidents and the characters of the plays.[168] Judge Hand in *Nichols* did not indicate that an abstraction analysis must begin from a single idea, but rather implies that more than one idea should be considered by the court.[169]

If a court finds the idea/expression boundary to be the most general idea abstraction—the point from which Judge Hand began the inquiry for the idea/expression boundary—the subsequent analysis will always grant overbroad copyright protection. Computer programs incorporate several ideas, many of which are not related to a general concept such as

---

162. *See, e.g.*, Hoehling v. Universal City Studios, 618 F.2d 972 (2d Cir.) (allegations of copying encompassed non-copyrightable material, finding summary judgment against plaintiff appropriate), *cert. denied*, 449 U.S. 841 (1980).

163. *Lotus*, 740 F. Supp. at 54; *see also* CONTU REPORT, *supra* note 28, at 20 (principle denying copyright protection to ideas extends to expression of ideas that have limited number of ways they can be expressed). "This principle [idea expression dichotomy] attempts to reconcile two competing social interests: rewarding an individual's creativity and effort while at the same time permitting the nation to enjoy the benefits and progress from use of the same subject matter." Sid & Marty Krofft Television Prods. v. McDonald's Corp., 562 F.2d 1157, 1163 (9th Cir. 1977).

164. Feist Publications v. Rural Tel. Serv. Co., 59 U.S.L.W. 4251, 4254 (Mar. 27, 1991) ("This Court has long recognized that the fact/expression dichotomy limits severely the scope of protection in fact-based works."); *see, e.g.*, Diamond v. Am-Law Publishing Corp., 745 F.2d 142, 148 (2d Cir. 1984) (informational work less protected than creative work); Landsberg v. Scrabble Crossword Game Players, Inc., 736 F.2d 485, 488 (9th Cir.) (factual works have narrower protection), *cert. denied*, 469 U.S. 1037 (1984).

165. 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931).

166. *See supra* note 46 and accompanying text.

167. *Nichols*, 45 F.2d at 122.

168. *Id.* at 121.

169. *See* Nimmer, Bernacchi & Frischling, *A Structured Approach to Analyzing the Substantial Similarity of Computer Software in Copyright Infringement Cases*, 20 ARIZ. ST. L.J. 625, 640 (1988) (abstractions test implicitly recognizes that work may consist of numerous ideas and expressions).

an electronic spreadsheet.[170] As an example, all programs need the ability to save completed work to a memory storage device.[171] This need calls for a choice from a menu selection.[172] Using the *Lotus* court's logic, if the menu command system is protectable expression, any increase in detail beyond that point is still protectable expression.[173] Where a menu choice is used merely to save work, the *Lotus* court would erroneously find an infringement under its stated logic.[174]

A computer program is more similar to a compilation of facts than a screenplay or novel. Facts themselves are not protected by copyright laws.[175] Nevertheless, the expression embodied in the compilation of factual information is protected,[176] as is any additional expression that the author may provide.[177] As such, factual works are treated differently from fictional works[178]—the scope of protection for factual works is more limited.[179] The "factual" elements of a computer program, such as the "Save" command, are the independent ideas found within many, if not all, programs. Following traditional factual analysis, with narrow

---

170. *See, e.g.*, Pearl Sys. v. Competition Elecs., 1988 Copyright L. Dec. (CCH) ¶ 26,338 (S.D. Fla. July 15, 1988) (involving copyright infringement action for competitive pistol-shooting timing device). The District Court for the Southern District of Florida did not stop its abstraction at a single idea, but recognized and identified the individual ideas that were expressed:

> At issue here are the two subroutines. The par time entry subroutine was designed to provide a method for the user to set a par time. That is the idea. The shot review subroutine was designed to allow the user to review the shots he or she has fired and to learn of the time that elapsed between each shot. That is also an idea. The subroutines themselves are expressions of those ideas.

Id.; *accord* Q-Co Indus. v. Hoffman, 625 F. Supp. 608, 615-16 (S.D.N.Y. 1985) (four of twelve modules of programs match—title, loading, text scrolling and character generator—but modules were similar in ideas rather than expression and were found not to infringe).

171. A memory storage device is a device that holds data, and from which the same data may be retrieved. MCGRAW-HILL DICTIONARY OF SCIENTIFIC AND TECHNICAL TERMS 922 (1974).

172. For example, in Lotus 1-2-3 the "Save" command is the menu choice that places data in the memory storage device. D. COBB & G. LEBLOND, *supra* note 70, at 233.

173. *See supra* notes 40-46 and accompanying text.

174. The court did not reach this level, having found infringement at the menu command level. *See supra* note 127.

175. *Feist Publications*, 59 U.S.L.W. at 4252.

176. 17 U.S.C. § 102 (1988).

177. Harper & Row Publishers v. Nation Enters., 471 U.S. 539, 556-57 (1985).

178. *Landsberg*, 736 F.2d at 488 (similarity requirement for infringement varies with type of work). "[Authors of factual works] often can choose from only a narrow range of expression. . . . [S]imilarity of expression may have to amount to verbatim reproduction or very close paraphrasing before a factual work will be deemed infringed." *Id.*

179. *See Harper & Row*, 471 U.S. at 563 ("The law generally recognizes a greater need to disseminate factual works than works of fiction or fantasy."); Narell v. Freeman, 872 F.2d 907, 914 (9th Cir. 1989) (more similarity allowed in informational work before finding of infringement than allowed in creative work).

protection for any expression of a fact, using the term "Save" would not be afforded any copyright protection.[180] Similarly, the term "Graph"—which is a "necessary fact" for electronic spreadsheet programs—would not be protected. The legal test the *Lotus* court implemented gave protection to these commands.[181]

Recognizing that programs express more than just one generalized idea will avoid the overbroad protection granted by the *Lotus* court. In *Manufacturers Technologies v. CAMS, Inc.*,[182] the court recognized the pitfalls of overbroad protection.[183] Considering a claim for infringement of a computer screen display, the court established the legal fiction of two separate copyright registrations, treating "the single registration of the computer program as accomplishing two interrelated yet distinct registrations; one of the program itself and one of the screen displays or user interface of that program, to the extent that each contains copyrightable subject matter."[184] Separate copyright registrations, the court reasoned, "[avoids] the mistake of identifying a program's idea with the idea of a particular screen display or some element therein. It recognizes that a computer program and its screen displays are, for copyright purposes, fundamentally distinct."[185]

B. *Program Routines Should Not Be Given Copyright Protection Just as* Scenes a Faire[186] *Are Excluded From Protection in Other Literary Works*

Along the lines of protecting programs in a fashion similar to factual works, courts should remain cognizant of the limitations placed upon computer programmers by the machinery for which they are writing programs. Unlike most artistic work that is expressed directly to another person, computer programs necessarily have a machine processing one person's expression for another person's perception.[187] This factor in-

180. This is in conformity with the doctrine of merger of ideas and expression as identified in the *CONTU Report*: "The 'idea-expression identity' exception provides that copyrighted language *may be copied* without infringing when there is but a limited number of ways to express a given idea." CONTU REPORT, *supra* note 28, at 20 (emphasis added).

181. *See supra* note 128 and accompanying text.

182. 706 F. Supp. 984 (D. Conn. 1989).

183. *Id.* at 995 (screen format conventions too narrow a range of possibilities to be afforded copyright protection).

184. *Id.* at 993.

185. *Id.*

186. "The French use a very expressive phrase in dramatic literature: 'scenes a faire'; that is, scenes which 'must' be done." Schwarz v. Universal Pictures Co., 85 F. Supp. 273, 275 (S.D. Cal. 1945).

187. This intervention, and more specifically the fact that a program must be converted to

troduces elements of utility and limitations to that expression with which courts must deal.

Copyright in literary works has long recognized that no protection will be extended to *scenes a faire*—those elements that are stock and trade of a story.[188] Similarly, computer programs have *scenes a faire* in "routines"[189] of program code that are standard for accomplishing a certain task on the computer.[190] There are limitations on the efficient use of a programming language by a programmer. It is therefore possible for a programmer to duplicate these routines in programs identically or nearly identically without intentional copying.[191] These routines are necessary expressions of an idea, and as such should not be protected expression.[192]

In computer programs, *scenes a faire* can be protected from direct copying. Programmers can, and in fact do, imbed identifiers within program code.[193] These identifiers—non-functional terms in the program code—evidence direct copying when found in an infringer's program. In

---

object code to enable a machine to utilize the program, has led to the observation that perhaps the object code should not be granted any copyright protection at all. Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 DUKE L.J. 663, 753.

188. *See, e.g.*, Walker v. Time Life Films, 784 F.2d 44, 50 (2d Cir.) (drunks, prostitutes, vermin and derelict cars are scenes that necessarily result from New York police story and are unprotectable "scenes a faire"), *cert. denied*, 476 U.S. 1159 (1986). "Under [the doctrine of *scenes a faire*], a second author does not infringe even if he [or she] reproduces verbatim the first author's expression, if that expression constitutes 'stock scenes or scenes that flow . . . necessarily from common unprotectable ideas,' because to hold otherwise would give the first author a monopoly on the commonplace ideas behind the *scenes a faire*." Landsberg v. Scrabble Crossword Game Players, Inc., 736 F.2d 485, 489 (9th Cir.) (quoting See v. Durang, 711 F.2d 141, 143 (9th Cir. 1983)), *cert. denied*, 469 U.S. 1037 (1984); *accord* Narell v. Freeman, 872 F.2d 907, 911 (9th Cir. 1989) ("[S]tock scenes containing little in the way of original expression are not protected.").

189. A routine is "part of a program, or a sequence of instructions called by a program, that may have some general or frequent use." J. ROSENBERG, *supra* note 10, at 454.

190. Data East USA v. Epyx, 862 F.2d 204, 208 (9th Cir. 1988) ("[No copyright protection granted to program] elements of expression that necessarily follow from an idea, or to 'scenes a faire'. . . ."); *see* Plains Cotton Coop. v. Goodpasture Computer Serv., 807 F.2d 1256, 1262 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987); Q-Co Indus. v. Hoffman, 625 F. Supp. 608, 616 (S.D.N.Y. 1985).

The application of this doctrine to computer programs had been discussed well before the decision in *Lotus*. *See, e.g.*, Note, Whelan Associates v. Jaslow Dental Laboratory: *Copyright Protection for the Structure and Sequence of Computer Programs*, 21 LOY. L.A.L. REV. 255, 298-99 (1987).

191. *Plains Cotton*, 807 F.2d at 1260-61.

192. *See, e.g.*, Frybarger v. IBM, 812 F.2d 525, 530 (9th Cir. 1987); Atari, Inc. v. North Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 616 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982).

193. *See, e.g.*, Apple Computer v. Franklin Computer Corp., 714 F.2d 1240, 1245 (3d Cir. 1983) (programmer name embedded in one program and word "applesoft" embedded in another found in defendant's program), *cert. dismissed*, 464 U.S. 1033 (1984); SAS Inst. v. S & H

this way, necessary expressions of common ideas are available to the public supporting the policy behind the copyright laws, yet slavish copying of another's work, even common routines, will remain protected.

If protection is to be limited in this manner, expert testimony is required for the trier of fact to recognize computer *scenes a faire*. The need for expert testimony has already been recognized by the courts. Expert testimony has been used for comparing the particular program at issue.[194] Expert testimony has also been used to show that substantially different expression can be created by a programmer from the same idea.[195] Indeed, as one court recognized, integrated tests of expert testimony and analytic dissection of programs "may well be the wave of the future."[196]

C. *Screen Display Infringement Should Require a Showing that the Screen Conveys Information or Expression Not Necessary To the Idea Being Conveyed*

One of the most obvious aspects of the user interface is the screen display. Courts have protected screen displays in some cases[197] and not protected them in others.[198] The decisive factor is whether the display contains expression.[199] The problem is determining at what point the screen display is more than just the conveyance of an idea.

A court should look to the screen displays and assess whether or not

---

Computer Sys., 605 F. Supp. 816, 824 (M.D. Tenn. 1985) (vestigial term from deleted operation in plaintiff's program found in defendant's program).

194. *See, e.g.*, E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485, 1497 (D. Minn. 1985). The court, in a technical holding which included a glossary of terms, compared several facets of the two programs made by experts and concluded that copied expression of the plaintiff's program was not the "only and essential means" of expressing the underlying idea. *Id.* at 1502.

195. *Id.* at 1493. In *Pearl Systems v. Competition Electronics*, a programmer was hired to write a timing program accomplishing the same goal as the defendant's program. 1988 Copyright L. Dec. (CCH) ¶ 26,338 (S.D. Fla. July 15, 1988). Without being informed of the plaintiff's program other than in the most general terms, the programmer independently developed a program that did not infringe on the plaintiff's program—a result the court considered when analyzing the "virtually identical" subroutines of the defendant's program. *Id.* at —.

196. Broderbund Software v. Unison World, 648 F. Supp. 1127, 1134 (N.D. Cal. 1986).

197. *See, e.g.*, Whelan Assocs. v. Jaslow Dental Laboratory, 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); Digital Communications Assocs. v. Softklone Distrib. Corp., 659 F. Supp. 449 (N.D. Ga. 1987); *Broderbund*, 648 F. Supp. 1127.

198. *See, e.g.*, Plains Cotton Coop. v. Goodpasture Computer Serv., 807 F.2d 1256 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987); Q-Co Indus. v. Hoffman, 625 F. Supp. 608 (S.D.N.Y. 1985).

199. Synercom Technology v. University Computing, 462 F. Supp. 1003, 1011 (N.D. Tex. 1978) ("The litmus seems to be whether the material proffered for copyright undertakes to express.").

they convey information that is not essential to the display itself. In *Manufacturers Technologies v. CAMS, Inc.*,[200] the court identified the idea behind the screen display[201] and then considered whether: (1) any expression existed in that screen, and (2) if the expression found was a necessary incident to that idea.[202] The plaintiff in *Manufacturers Technologies* had created a program for calculating a cost estimation of machining a part.[203] An element of this program was an on-screen display of the user's prior selections while running an estimation.[204] While the "idea" of displaying prior selections or status was not copyrightable, the expression, or selection and placement of what information would be shown on the display, was not a necessary incident and thus could be copyrightable.[205] The court in *Manufacturers Technologies* held that a screen prompting for information must also convey information for a finding of copyrightable expression.[206] Limiting the breadth of protection of computer programs as a result of hardware constraints, the court observed that the "screen page is only so long and so wide" and therefore placement on the display screen is necessarily limited.[207] The court explicitly denied protection to the use of particular terms or symbols on the screen.[208]

Having defined the expression, a court must then find the expression can be separated from the idea being expressed.[209] Indispensable expression should be protected only from virtually identical copying.[210] A court should consider whether the expression—or structure, sequence and layout—of the screens is dictated by artistic or functional considera-

200. 706 F. Supp. 984 (D. Conn. 1989).

201. The idea was to apprise the user of the status of his or her efforts during the cost-estimation of the machining of a part. *Id.* at 996.

202. *Id.*

203. *Id.* at 988.

204. *Id.* at 996. The program identified the operation or department utilized, the tooling used in the department, and the type of material being used. *Id.*

205. *Id.* The court found, "That expression reflects selection as to what should be made part of the status report, arrangement of the terms therein, assignment of numbers to specific operations/departments and tools, and coordination in the manner of building on the status report as the user progresses through various steps." *Id.*

206. *Id.* at 997; *see also Synercom*, 462 F. Supp. at 1012 (formats may by placement of lines, shaded art, and words create copyrightable expression telling user what, where and how to place data).

207. *Manufacturers Technologies*, 706 F. Supp. at 995.

208. *Id.* at 996 n.16.

209. *See* Frybarger v. IBM, 812 F.2d 525, 530 (9th Cir. 1987) (expression of idea based on technical requirements of program is indispensable expression); *Synercom*, 462 F. Supp. at 1012 ("[F]ormats are copyrightable if the ideas they express are separable from their expression.").

210. *Frybarger*, 812 F.2d at 530.

tions. For example, choosing a typeface for display on the screen or particular wording of phrases expressing commands would ordinarily be dictated by artistic or aesthetic factors and receive protection.[211] On the other hand, an alphabetic, two column display is a functional consideration and should not receive protection.[212]

The court in *Lotus* failed to make a proper assessment of the screen displays. In reviewing the menu command system, the *Lotus* court found that a two-line moving cursor display was one of many ways of expressing a menu system.[213] This alone should not be a sufficient criteria for determining copyrightability in view of the limited number of ways of displaying a menu system.[214] It appears from the holding of the *Lotus* court that Paperback had infringed upon the expression of the 1-2-3 menu by its near verbatim copying of the phrases displayed on the screen.[215] This finding of infringement could properly be made without unnecessarily granting a copyright monopoly to the first programmer using a two-line cursor.

D. Courts Should Consider Factors in the Marketplace, in Limited Instances

There is an important reason why computer programs should not be compared to a stage play when searching for the extraction of expression from idea.[216] Enjoyment of one play is independent of other plays—it is difficult to define a "learning curve" for the public to enjoy a play. Computer programs, however, are unique among the works afforded copyright protection because they require some degree of "learning" before utilization.

The consumer's greatest investment in computer software is not in the software itself, but rather in the time and money involved in learning

211. Broderbund Software v. Unison World, 648 F. Supp. 1127, 1134 (N.D. Cal. 1986) (court followed such analysis in clearly defining expression and idea).

212. *Manufacturers Technologies*, 706 F. Supp. at 996.

213. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 65 (D. Mass. 1990).

214. The *Lotus* court found a two-line cursor and the spelling of menu commands to be original expression of a menu command system. *See supra* notes 127-30 and accompanying text. More appropriate is the finding of *Manufacturers Technologies* that "[t]he use of columnar format and the use of both upper and lower case letters are not sufficient on their own to warrant copyright protection because they lack originality." *Manufacturers Technologies*, 706 F. Supp. at 998; *cf. Digital Communications*, 659 F. Supp. at 459 (1987) (use of command driven program—typing of two symbols or letters—non-protectable idea yet capitalization or highlighting of those letters is protected expression).

215. *Lotus*, 740 F. Supp. at 86-87.

216. *See supra* notes 161-70 and accompanying text.

to use it properly.[217] Protecting the user interface of a program, particularly after it has become a "standard," can block the continued work of other programmers.[218] Consumers would be unwilling to endure, again, the time and expense involved in complete retraining for the use of a new program—in effect, the public would be "check-mated" by the original copyright holder.[219]

Copyright does not absolutely preclude an author from referring to and relying upon prior published material.[220] In fact, courts must occasionally subordinate the interest of the copyright holder to that of the public in order to meet the constitutional purpose of copyright protection—development of art, science and industry.[221] If it appears that granting protection to the expression could exhaust all possibilities of future use of the subject matter, copyright protection should not be granted.[222]

The legal test and analysis used by the *Lotus* court overlooks this overriding consideration. Concern over the breadth of the *Lotus* holding has been expressed within the computer industry.[223] Software programmers, the very population that allegedly is being protected, have formed at least one organization to oppose copyright protection of software in light of the court's decisions.[224] The *Lotus* court went to great lengths to support its dismissal of Paperback's argument that programs are "built" upon each other, and those "building blocks" should not be afforded copyright protection.[225] Historically, both case precedent and the

217. Currid, *Training Won't Cure a Case of Bad Technology*, PC WEEK, Mar. 18, 1991, at 70.

218. A common user interface will benefit both users, allowing them to learn only one set of commands, and programmers, allowing them to focus on innovative developments. Kay, *Group Protests Copyright Law*, LAN TIMES, Sept. 1990, at 40.

219. Ironically, the newest release of Lotus 1-2-3 (Release 3.1 (1990)) utilizes another program to allow for capabilities that are popular in the marketplace. One reviewer noted: "[T]here's one hitch. You now have two separate menu systems to master [when utilizing the new Lotus 1-2-3 program.]" Stinson, *First Looks*, PC MAG., Nov. 13, 1990, at 33, 34.

220. Rosemont Enters. v. Random House, Inc., 366 F.2d 303, 310 (2d Cir. 1966) ("It is just such wasted effort that the proscription against the copyright of ideas and facts, and to a lesser extent the privilege of fair use, are designed to prevent."), *cert. denied*, 385 U.S. 1009 (1967).

221. *Id.* at 307.

222. *See* Morrissey v. Procter & Gamble, 379 F.2d 675, 678-79 (1st Cir. 1967) ("We cannot recognize copyright as a game of chess in which the public can be check-mated.").

223. *See, e.g.*, Dvorak, *The Death of Code*, PC MAG., Nov. 13, 1990, at 81; *Litigious Software Industry Needs Taming*, PC WEEK, July 16, 1990, at 55; Seymour, *Lotus vs. Borland Upshot: Customer Intimidation?*, PC WEEK, July 23, 1990, at 15.

224. *See* Kay, *supra* note 218, at 40.

225. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990) (court discussed this under the section of the "OTSOG" principle—"On The Shoulders Of Giants"). Contrary to the dismissal of this concept with regard to Paperback's program expressed at this

consideration of this issue by the CONTU Commission indicates the *Lotus* court was incorrect.[226] Elements of the user interface should be analyzed individually for protectable elements of expression. A list of contents or ingredients is not subject to copyright protection—they are necessarily incident to the idea that they express and are dictated by those functional considerations.[227] Yet the court granted protection to 1-2-3's menu command structure because it considered functional choices the user would make and was not just based on the alphabet as was VisiCalc's menu.[228] The Lotus 1-2-3 menu/screen displays do not provide information; to the contrary, they are arranged in a fashion *dictated by function*[229] and should not have been afforded protection.

VI. CONCLUSION

Congress should adopt the Commission's suggestion that software copyright protection be reviewed periodically.[230] An evaluation of the protection currently provided by a re-convened Commission would allow a detailed consideration of judicial rulings and legislative amendments to the 1976 Copyright Act that affect software copyright. The results of such an analysis can then be used by the legislature to decide whether their goals are being achieved, or if the subject matter has created issues that can best be solved through a *suis generis* statute.[231]

New authors of computer software should be satisfied that their work will be protected, but the balance between the protection of expression and encouragement of new works is extremely delicate. The legislature, in its most recent consideration of computer software copyright,[232] recognized the important position software occupies with today's pub-

---

portion of the opinion, the court previously stated that elements of Lotus' program, 1-2-3, "could thus be thought of as an evolutionary product that was built upon the shoulders of VisiCalc." *Id.* at 66.

226. *See supra* notes 20-69 and accompanying text.

227. Manufacturers Technologies v. CAMS, Inc., 706 F. Supp. 984, 996 (D. Conn. 1989).

228. *See supra* notes 127-30 and accompanying text.

229. The court found the submenus presented "a list of up to about ten full-word menu choices, presented in order of predicted frequency of use rather than alphabetically." *Lotus*, 740 F. Supp. at 67.

230. *See supra* note 39 and accompanying text.

231. In developing a new copyright law, China looked to the United States copyright statute as a model for much of its new statute. Sobel, *Panel 4: Technological Transfer and Protection of Intellectual Property in China*, 12 LOY. L.A. INT'L & COMP. L.J. 61, 66 (1989). Finding this law inadequate for the protection of software, China departed from the United States "model" and created a separate statute for software protection. *Id.* at 66-67.

232. Computer Software Rental Amendments Act of 1990, Pub. L. No. 101-650, §§ 801-805, 104 Stat. 5089, 5134 (1990) (to be codified as amended at 17 U.S.C. § 109).

lic.[233] The legislature also recognized the intrinsic utilitarian purpose of software[234] and the desire of some authors to allow, with minor constraints, the use of their original programs.[235]

Through a misapplication of law, the *Lotus* court found that Paperback Software violated the copyright on Lotus' 1-2-3 program. In so doing, the court failed to recognize several factors unique to computer software and granted copyright protection to the program's "user interface" without identifying protectable expression.

If the courts: (1) recognize that programs express several ideas, (2) rule out non-protectable "routine" or standard elements of programming from the remaining expression, and (3) consider the implications for future program creation the grant of a copyright monopoly will have, then new authors will properly be protected, new work will be encouraged, and it will be easier for programmers to recognize what may be an infringement of other programs without litigating each program.

Courts for the most part have done an admirable job applying legal paradigms—grounded in the arts—to a highly technologic field. However, periodic review by commissions representative of each affected area will keep derailing opinions such as *Lotus Development Corp. v. Paperback Software International*[236] on track. Until such time, this opinion will hopefully remain a unique aberration from the proper scope of computer software copyright.

Craig A. Laidig

233. "Computers have become commonplace in government, our homes and offices, and business enterprises. Software—the technology that makes computers work—is of pivotal importance to the United States, which is the world's leader in this unique form of creativity." H.R. REP. No. 735, 101st Cong., 2d Sess. 4, *reprinted in* 1990 U.S. CODE CONG. & ADMIN. NEWS 6935, 6935.

234. *Id.* at 6939.

235. *Id.* at 6947-48.

236. 740 F. Supp. 37 (D. Mass. 1990).