January 1992

Look and Feel: A Proposed Solution to the Diverging Views between the Software Industry and the Courts

Walter G. Duflock

Follow this and additional works at: http://digitalcommons.law.scu.edu/chtlj

Part of the Law Commons

Recommended Citation
Available at: http://digitalcommons.law.scu.edu/chtlj/vol8/iss2/6

This Comment is brought to you for free and open access by the Journals at Santa Clara Law Digital Commons. It has been accepted for inclusion in Santa Clara High Technology Law Journal by an authorized administrator of Santa Clara Law Digital Commons. For more information, please contact sculawlibrarian@gmail.com.
"LOOK AND FEEL": A PROPOSED SOLUTION TO THE DIVERGING VIEWS BETWEEN THE SOFTWARE INDUSTRY AND THE COURTS

Walter G. Duflock†

Table of Contents

Introduction ........................................... 448

I. Current Protection .............................. 449
   A. Protection of non-literal aspects of computer software is unclear under the Federal Copyright Act ............................................ 449
   B. Federal court decisions are expanding copyright protection for non-literal elements of computer software .............................................. 450
      1. Courts are split regarding the proper scope of copyright protection for structure, sequence, and organization. .............................. 451
      2. Lotus v. Paperback: a recent case establishes a three-part test for determining infringement of non-literal software elements ..................... 453
      3. Lotus v. Borland: the next case illustrates why differentiation between a clone and an industry standard is required .......................... 455

II. Computer Programmers and Computer Users Both Oppose Extending Copyright Protection to Non-literal Software Elements .............. 457
   A. Industry experts believe copyright protection of non-literal elements of computer software is not required to promote innovation ..................... 458
   B. Expanding copyright protection contradicts users' desire for standardized software .................. 460

Copyright © 1992 by Walter G. Duflock.
† J.D. 1992, Santa Clara University School of Law; B.A. 1988, California Polytechnic State University, San Luis Obispo.
INTRODUCTION

The appropriate extent of copyright protection for the "look and feel" of computer software has generated substantial litigation and commentary.¹ The "look and feel" controversy involves graphical user interfaces (GUIs, pronounced "gooeys"), which are visual representations of software on a computer screen. Common examples of GUIs include the Macintosh operating system (pull-down menus and icons) and applications such as Lotus 1-2-3 (two-line menu system of commands and L-shaped spreadsheet) and Windows (which creates a Macintosh-like interface on IBM PCs).² The extent of copyright protection for GUIs is currently a fundamental conflict in computer law.

Most GUI litigation involves software companies claiming a competitor's GUI is an infringement on their product's copyright. For example, Lotus claims that the GUIs of two competitors - Borland's Quattro and Paperback Software's VP-Planner - infringe on the Lotus 1-2-3 copyright. In addition, Apple claims Microsoft

---

¹. See, e.g., Peter Menell, Note, Defining the Scope of Copyright Protection for Computer Software, 41 STAN. L. REV. 497, 500 (1986); Diane Brinson, Copyrighted Software: Separating the Protected Expression from Unprotected Ideas, A Starting Point, 29 B.C. L. REV. 803 (1988).

². Paul Nesbitt, New Wave optimism; Hewlett-Packard Co.'s NewWave graphical environment, PC USER, August 29, 1990, at 32.
Windows infringes on Apple's System and Finder copyright. In many instances, the alleged infringer is either (1) a "clone" designed to steal market share by selling at a lower price or (2) an industry advancement which builds on another product's strengths and improves on the first software product. Differentiating between these types of infringements is crucial. Public policy requires that copyrighted software be protected against "clones," which discourage innovation by limiting the inventor's return, but not be protected against industry advancements, which are a further innovation.

This comment will address the appropriate scope of copyright protection for GUIs. The first section analyzes the current judicial expansion of copyright protection. The second section reviews computer programmers' and users' beliefs that extending copyright protection for computer software will hurt innovation and prevent standardization. The third section offers a proposed legislative solution to the conflict between programmers and users, who oppose increased protection, and the courts, which are increasing protection. By clarifying the scope of copyright protection for computer software, the proposed legislation will insure that innovation, not litigation, dominates the computer software industry during the next decade.

I. CURRENT PROTECTION

A. Protection of non-literal aspects of computer software is unclear under the Federal Copyright Act.

Copyright protection for computer software began with the Computer Software Copyright Act of 1980.© CONTU characterized software as a relatively "new type of writing" and concluded that copyright protection was essential to provide incentive for the development and widespread dissemination of computer software.©

Unfortunately, this amendment does not resolve the idea/expression dichotomy regarding computer software. Copyright law

---


4. CONTU at 1-11.
does not protect ideas. However, copyright law does protect individual expressions of ideas, provided they are "original works of authorship fixed in a tangible medium of expression." Drawing the line between unprotectable ideas and protectable expression of ideas in computer software has become increasingly more difficult. Recently, the courts have begun to distinguish between literal and non-literal aspects of computer programs.

Literal components include source code and object code. Currently, copyright protection is available for source code and object code. The protection of non-literal components under the Copyright Act is unclear. Non-literal software components of a computer program include the sequence, structure, and organization of the program, as well as the screen output or user-interface (the "look and feel" of the program). The 1980 Copyright Act amendments were written before controversy existed over differing protection for literal and non-literal software elements. As a result, the amendment does not specifically mention whether non-literal software elements such as GUIs were intended to be protected or unprotected. Consequently, federal courts and commentators have spent the past decade trying to decide how much protection non-literal aspects of computer programs should be entitled to under the Copyright Act.

B. Federal court decisions are expanding copyright protection for non-literal elements of computer software.

There is no consensus among the federal circuit courts on the

7. Source code is code that is readily understandable by the programmer. Programs are usually written in source code then translated by a "compiler" program into object code for use by the computer. Thus, object code and source code are really two different versions of the same program. Apple Computer Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983).
8. Object code is code that is understandable to the computer. Programs are converted from source code to object code by a compiler, which takes the program from human-readable input (source code) and converts it into machine-readable input (object code). Id. at 1253. See generally Stern Elec., Inc. v. Kaufman, 669 F.2d 852, 855 n.3 (2d Cir. 1982) (written programs are copyrightable as literary works); Williams Elec. v. Arctic International, 685 F.2d 870, 876-77 (3d Cir. 1982) (object code copyrightable); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983) (source and object code copyrightable).
11. See, e.g., Menell, supra note 1, at 1045 (suggests an economic framework for deciding the appropriate scope of protection); Brinson, supra note 3, at 853 (advocates decisions should be made case-by-case based on a "level of abstractions test").
appropriate extent of copyright protection for non-literal aspects of software. Some courts allow protection of structure, sequence, and organization. Other courts have refused to extend copyright protection to non-literal software elements, holding that they are ideas, not expressions, and therefore not entitled to copyright protection.

1. Courts are split regarding the proper scope of copyright protection for structure, sequence, and organization.

The principal case allowing copyright protection for structure, sequence, and organization is a Third Circuit case, Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.\(^\text{12}\) Whelan designed a computerized accounting system for Jaslow.\(^\text{13}\) Disputes arose over a contract specifying a profit-sharing plan for proceeds from the software.\(^\text{14}\) Jaslow broke the contract and developed a competing computerized accounting system for dental laboratories. Whelan sued, claiming Jaslow's software infringed Whelan's software copyright.

Whelan did not claim that Jaslow copied the literal program code in Whelan's software. Instead, Whelan claimed that Jaslow's program copied the structure, sequence, and organization of Whelan's program. At trial, expert testimony revealed that similarities in file structure, screen outputs, and overall structure were evident. The court used this evidence to hold that "copyright protection of computer programs could extend beyond the programs' literal code to their structure, sequence, and organization. . . ."\(^\text{15}\)

Whelan established a very loose standard for copyright protection. Whelan defined the underlying purpose of a program as its idea, and held that everything else was expression.\(^\text{16}\) In Whelan, the idea of the program was the "efficient management of a dental laboratory." Because that idea could be accomplished in a number of different ways, the structure of the program is part of the program's expression, not its idea, and is therefore protectible by copyright.\(^\text{17}\) Under Whelan, the "purpose" of the software program, as determined by the court on a case-by-case basis, may be determined at

---

13. Id. at 1225-26.
14. Id. at 1223.
15. Id. at 1222.
17. Whelan v. Jaslow, 797 F.2d 1222 at 1238 n.28.
such a high level of abstraction that virtually any elements of program structure, sequence, or organization are protectible.\textsuperscript{18}

The Fifth Circuit in \emph{Plains Cotton Cooperative Association v. Goodpasture Computer Service, Inc.},\textsuperscript{19} took the exact opposite position. Plains Cotton developed a computerized cotton marketing program that allowed a cotton producer to track a bale of cotton from the gin to a seller and electronically account for the sale.\textsuperscript{20} Former employees of Plains Cotton began working for a competitor and completed the design of a competing system within weeks.\textsuperscript{21} Plains Cotton sued, claiming the competing product infringed on Plains' software copyright because the "input formats" (data entry screens) were similar.\textsuperscript{22} As in Whelan, the plaintiff in Plains Cotton did not allege copying of program code. The court relied on Judge Higginbotham's reasoning in Synercom Technology, Inc. v. University Computing Co.\textsuperscript{23} The court specifically declined to embrace Whelan, instead holding that input formats were ideas, not expressions, and thus were not protected by copyright.\textsuperscript{24}

\textit{Plains Cotton} and \textit{Synercom} hold that copyright protection for software does not extend to structure, sequence, and organization because they constitute an unprotectible idea, not protectible expression. However, other cases have followed the \textit{Whelan} approach and extended copyright protection to non-literal elements of software. For example, courts have held that the overall structure of a computer program,\textsuperscript{25} as well as screen displays,\textsuperscript{26} user interfaces,\textsuperscript{27} and sequence and organization\textsuperscript{28} are valid subjects for copyright protection. Thus, many courts continue to expand copyright protection for computer software.

\begin{footnotes}
\item[18] In Whelan, the court defined the "purpose" of the program as "run[ning] a dental laboratory in an efficient way." \textit{Id.} at 1238 n.34. Under this definition, almost any organization feature of the program may be protected by copyright.
\item[20] \textit{Id.} at 1258.
\item[21] \textit{Id.} at 1258-59.
\item[22] \textit{Id.} at 1262.
\item[28] \textit{Id.}
\end{footnotes}

*Lotus Development Corp. v. Paperback Software Ltd.*, 29 is the most recent judicial attempt to draw the line between unprotected ideas and protected expression. Lotus developed Lotus 1-2-3, the dominant spreadsheet for use on IBM personal computers and IBM-compatible clones. Paperback Software later developed VP-Planner, a spreadsheet that competed as a low-cost Lotus 1-2-3 look-alike program. Lotus sued Paperback, claiming that VP-Planner infringed on the copyright of Lotus 1-2-3.

The *Lotus v. Paperback* court established a three-prong test which decides the scope of copyright protection for software on a case-by-case basis. 30 The test is summarized as follows: (1) the court must define and locate the “idea” of the software along a sliding scale that runs from generalized ideas to progressively more particularized expression; (2) the decision-maker must decide whether the alleged expression is one of a limited number of ways of expressing the idea or whether there are other ways to express the same idea; and (3) if the decision-maker finds there are other ways to express the same idea, he must then determine whether the allegedly copyrightable work is a substantial part of the work. 31 If all three parts are met, the plaintiff has proven his infringement case and the defendant is liable for copyright infringement.

For the first prong of the test, the court must locate where the copyrighted item’s structure falls on the idea-expression continuum of available alternative expressions. Such a decision involves looking at the facts of each case and therefore requires a balancing test. 32 Judge Keeton stated that it is wiser not to “press the search for a suitable bright-line test of copyrightability where [Judge] Learned Hand, even after decades of experience in judging, found none.” 33 The closer the structure is to expression, the more protection it will be given. Consequently, the more specific the structure, the more likely the court will allow copyright protection. Judge Keeton found that the command structure and menu hierarchy of Lotus 1-2-3 were capable of being expressed in alternate ways; thus,

30. *Id.* at 59-62.
33. *Id.* at 60.
the command structure was not "obvious" and was validly copyrightable.\textsuperscript{34}

Second, the court must decide which elements of expression are not essential to every expression of the idea.\textsuperscript{35} The court considers the particular expression in terms of all possible expressions. The more expressions that are possible, the more likely the court is to grant protection to this particular expression. In \textit{Lotus v. Paperback}, both (a) the rotated "L" indicating the rows and columns of the computer spreadsheet,\textsuperscript{36} and (b) the two line, moving cursor menu invoked by the slash key\textsuperscript{37} were found to be essential to computer spreadsheet programs and therefore not protected by copyright.\textsuperscript{38} However, both (a) the assignment of particular function keys to frequently-used commands such as Help and Edit and (b) the "macro" feature used to automate sequences of repetitive commands and keystrokes were found not to be essential to spreadsheets and were therefore properly protected by copyright.\textsuperscript{39}

Third, the court must decide whether the protectible elements of expression are a substantial part of the allegedly copyrightable work.\textsuperscript{40} The result is that non-literal elements of software are protectable by copyright if there are other possible ways to express the idea, and the copyrightable elements of expression are a substantial part of the work. In \textit{Lotus v. Paperback}, Paperback's assignment of identical function keys and macro features was determined to be a substantial part of VP-Planner; therefore, VP-Planner infringed on the copyright of Lotus 1-2-3.

In \textit{Lotus v. Paperback}, Judge Keeton found that under the first prong of the test the command structure of Lotus 1-2-3 was not obvious and therefore was protectible by copyright. Under the second prong, both the assignment of particular function keys to frequently-used commands and the macro feature were not essential to

\begin{thebibliography}{9}
\bibitem{34} D. Lee Antkon & Gary M. Hoffman, Copyright Protection and Innovation: The Impact of Lotus Development v. Paperback Software 7 \textsc{The Computer Law} 1, 3 (August 1990).
\bibitem{35} \textit{Lotus v. Paperback}, 740 F. Supp. at 61.
\bibitem{36} The visual element of the user interface mimics the rows and columns found in a printed spreadsheet. Lewis, \textit{supra} note 31, at 697.
\bibitem{37} The menu contains commands a user selects to carry out common tasks such as copying data in a spreadsheet. These commands and their arrangement were nearly identical in the VP-Planner and Lotus 1-2-3 user interfaces. \textit{Lotus v. Paperback}, 740 F. Supp. at 70.
\bibitem{38} \textit{Id.} at 66.
\bibitem{39} Just as a tape recorder lets a person record and play back sound, the macro feature of Lotus 1-2-3 and VP-Planner lets a user record sequences of keystrokes and then "play back" the keystrokes so as to automate many repetitive tasks. Lewis, \textit{supra} note 31, at 697-98.
\bibitem{40} \textit{Lotus v. Paperback}, 740 F. Supp. at 61.
\end{thebibliography}
the expression of a spreadsheet. Therefore, these two elements were properly protected by copyright. Under the third prong, VP-Planner was held to infringe Lotus 1-2-3 because Paperback's use of the two copyrighted elements was a substantial part of VP-Planner.

*Lotus v. Paperback* will not be appealed. Paperback agreed to an out-of-court settlement which required Paperback to: (1) pay Lotus $500,000; (2) stop marketing its spreadsheet products V-P Planner and V-P Planner Plus; (3) not appeal Judge Keeton's June ruling; and (4) drop its counterclaim against Lotus. Therefore, its precedential value is fairly small. Specifically, the District Court judgment will have little direct impact on future cases such as *Lotus v. Borland* and the Apple-Microsoft litigation. However, *Lotus v. Paperback* continues the federal court's expansion of copyright protection to non-literal aspects of computer software. If the analysis in *Lotus v. Paperback* is applied to *Lotus v. Borland*, an industry advancement will be held an infringement.

3. **Lotus v. Borland**: the next case illustrates why differentiation between a clone and an industry standard is required.

An analysis of the facts in *Lotus v. Borland* and *Lotus v. Paperback* will show that the two cases are factually different. Paperback's program, VP-Planner, was advertised as a lower-priced Lotus 1-2-3 "compatible" program which would use the same commands and menus as 1-2-3. Compatibility would allow users to (1) switch from 1-2-3 to VP-Planner without retraining and (2) use 1-2-3 spreadsheet files without losing features or flexibility. VP-Planner offered only one menu choice, the Lotus 1-2-3 look-alike menu.

However, Borland's program, Quattro, is not a cheap imitation of Lotus 1-2-3. Quattro offers the 1-2-3 menu as one of three menu options; Quattro also allows users to choose between (1) using Quattro's own menu system or (2) creating a customized menu system. Like VP-Planner, Quattro offers the 1-2-3 menu as an option so that Lotus users will upgrade to a competitor's product. However, unlike VP-Planner, which advertised itself as a low-cost 1-2-3 compatible spreadsheet, Borland advertises Quattro as a product superior to Lotus 1-2-3. Software product reviews have acknowl-

---

42. *Id.*
43. Lewis, *supra* note 31, at 697.
44. Borland's advertisements in industry magazines such as InfoWorld and PC Week, as well as the Wall Street Journal and New York Times ask consumers to "Judge for yourself. Quattro Pro is radically different from Lotus 1-2-3. At Borland we believe in software crafts-
edged Quattro as a superior product.45

In addition, the motivations for litigation are different. VP-Planner was advertised as, and purchased by consumers as, a low-cost 1-2-3 “compatible” spreadsheet clone. VP-Planner infringed Lotus 1-2-3 because it was a 1-2-3 clone that did not advance the state of the art. Lotus sued Paperback to prevent Paperback, an intentional low-priced clone, from stealing the market share of Lotus 1-2-3.

However, Quattro was chosen by users over Lotus 1-2-3 based on two factors: (1) Quattro was cheaper than Lotus 1-2-3,46 and (2) Quattro was a better spreadsheet than Lotus 1-2-3.47 Lotus sued Borland to prevent an industry advancement from legitimately gaining market share in the spreadsheet market. In fact, it has even been suggested that one reason Lotus initiated the lawsuits against Borland and Paperback Software was to increase market share. One analyst believes the Lotus lawsuits have cost Lotus “a lot of goodwill in and out of the industry.”48

Users purchased VP-Planner as a low-priced 1-2-3 compatible spreadsheet. Users purchased Quattro as a low-priced 1-2-3 compatible spreadsheet that is superior to 1-2-3 and advances the state

45. See, e.g., T.R. Reid and Brit Hume, Upgraded Spreadsheet a Real Pro, CHI. TRIB., July 22, 1990, at C10. Reid and Hume write that Quattro Pro is an upgraded spreadsheet that Lotus can’t beat on “price, quality, innovation or service.” Id.

46. Lotus 1-2-3 is available for $345 and Quattro Pro (an upgraded version of Quattro) is available for $282. MICROTIMES, December 10, 1990, at 4. Quattro has less features and costs less than Quattro Pro. When asked about Quattro, one user at Hughes Aircraft said, “We bought Quattro because of the price - it costs one-fourth of what Lotus does.” Neil Margolis, Users biggest losers in spreadsheet wars, COMPUTERWORLD, July 16, 1990, at 8. Quattro’s cheaper price was largely due to programming efficiencies that occurred since the initial development of Lotus 1-2-3, not from copying Lotus 1-2-3’s program code. This is a valid justification for lower price.

47. “If you can’t beat the competition on price, quality, innovation or service, you can always fall back on the last resort: Hire a lawyer and go to court. That’s what they teach in the business schools nowadays, and that’s the lesson Lotus remembered when it ran up against tough competition from Borland’s admirable new spreadsheet, Quattro Pro.” Reid and Hume, supra note 45, at 10.

of the art for spreadsheets. VP-Planner is a clone; Quattro is an industry advancement.

The ability of both judges and legislators to differentiate between these types of programs will determine whether innovation or litigation dominates the software industry in the next five years. Judges and legislators can improve their understanding of the issues involved in deciding the appropriate level of copyright protection for computer software by looking to the views of: (1) the computer industry, the innovators copyright protection is intended for; and (2) software users, whose general welfare copyright protection is supposed to protect.

II. COMPUTER PROGRAMMERS AND COMPUTER USERS BOTH OPPOSE EXTENDING COPYRIGHT PROTECTION TO NON-LITERAL SOFTWARE ELEMENTS.

Copyright law provides an incentive for innovation by allowing inventors to prevent others from infringing material protected by copyright. "The ultimate goal of copyright law is to stimulate creativity for the public welfare by securing a fair return for a creative labor." Therefore, the views of software programmers and users must be considered when determining the appropriate scope of copyright protection for computer software. This section discusses: (1) the level of protection industry experts feel is required to promote innovation without stifling creativity and (2) users' opinions relevant for determining the level of protection that will maximize the public (i.e. software users') welfare.

49. Recent events confirm Quattro's status. Borland has announced a "maintenance release" version 3.01 of Quattro Pro (the successor to the original Quattro) to maintain its superiority in product quality. Version 3.01 accelerates the WYSIWYG (what you see is what you get) graphic screen functions, pull-down menus, and mouse control. Owners of 3.0 who request the revision from Borland can get it for free. Lotus has announced 1-2-3 Release 2.3, what it calls a "major upgrade," in an attempt to remain competitive with Quattro Pro. Lotus' 1-2-3 upgrade includes many features that account for the success of Quattro Pro, including WYSIWYG features, graphic display and printing, interactive dialog boxes, and "vastly improved" help facilities. However, Borland's prices continue to beat Lotus. Proving that it will remain price-competitive, Borland is allowing users of Quattro Pro version 3.0 to upgrade to 3.01 for free, while Borland charges $59 for its alleged major upgrade to version 2.3. Cairn MacGregor, Users will end up paying for software litigation, THE GAZETTE (Montreal), July 10, 1991, at F5.

50. Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975).
A. Industry experts believe copyright protection of non-literal elements of computer software is not required to promote innovation.

The primary arguments supporting copyright protection of non-literal elements of computer software are (1) promoting innovation of software developers and (2) minimizing risk to software developers by providing copyright protection for design of non-literal software elements. David Reed, the chief scientist of spreadsheets at Lotus Development Corporation, feels it is fair that software designers who carefully create a useful organization of screens and menu choices should be able to prevent unlicensed use of their design.\(^{51}\) Reed believes much of the value of a good software product is in the user interface design.\(^{52}\) Additionally, Reed believes small software companies would take an enormous risk in introducing products if copyright protection for user interfaces were not available.\(^{53}\)

However, the better view is that copyright protection should not be extended to non-literal software elements. Dan Bricklin and Mitch Kapor, developers of two major computer spreadsheet programs,\(^{54}\) spoke to the House of Representatives Panel on Software Protection.\(^{55}\) Bricklin told the Panel that confusion is high among software developers because of recent court decisions and stated that "copyright protection for the object code is the bedrock of software protection."\(^{56}\) In Bricklin's opinion, trade secret and copyright protection of program code and trademark protection of software program titles is sufficient to promote innovation in the software industry.\(^{57}\) The benefits of further copyright protection are outweighed by the added costs of litigation and uncertainty this protection creates in the software industry.

Mitch Kapor believes less protection may be more effective in promoting innovation. Kapor told the House Panel on Software Protection that "over-protection of intellectual property is as perni-
cious as under-protection in its stifling effects on innovation." Kapor does not feel copyright protection should be extended beyond "literal expression," or source code. Kapor dislikes copyright protection of non-literal aspects of software because it creates uncertainty among commercial software companies as to what products can be created without fear of litigation.

A 1989 industry survey confirms that the Bricklin/Kapor view is the majority view in the computer software industry. Pamela Samuelson and Robert Glushko asked industry experts about the issue of copyright protection at the Association for Computing Machinery Conference on Computer-Human Interaction (CHI), held in May of 1989. The CHI meeting included most major computer companies, including many of the litigants in the "look and feel" area.

The survey concluded that computer programmers overwhelmingly support copyright protection for literal software elements such as source and object code. However, programmers strongly oppose copyright protection for non-literal elements such as GUIs. This is an unexpected result, since many of the respondents were responsible for creating the most commercially valuable user interfaces in the software industry. In spite of this, respondents strongly believed copyright protection should not extend to GUIs and other non-literal aspects of computer software.

The respondents believed strong copyright protection for non-literal elements would have a clear negative effect both on the industry and on their own work. Over seventy percent of the respondents expected a negative impact on their work if the current lawsuits established strong copyright protection, while only nine percent expected the effect on their work to be positive. Obviously, software industry innovators feel innovation will occur without copyright protection of non-literal aspects of computer

---

58. Id.
59. Id.
60. The CHI conference is the largest gathering of user interface researchers, designers, and developers - the people who have the most to gain or lose by the outcome of the look and feel controversy. Pamela Samuelson and Robert Glushko, ACM Conference: Survey on Computer-Human Interaction, 33 ASS'N FOR COMPUTING MACHINERY 483 (May 1990).
61. Representatives from Ashton-Tate, Lotus, Microsoft, IBM, Hewlett-Packard, Apple, MCC, AT&T, Texas Instruments, Xerox, Bell Communications Research, the University of Michigan, and Carnegie-Mellon University attended. Id.
62. Fully 93% of those with an opinion supported intellectual property protection for source code. In addition, 85% supported object code protection. Id.
63. Id.
64. Id.
software. Software industry experts believe copyright should not be extended beyond source code because the resulting litigation and uncertainty about the scope of copyright protection outweigh the benefit of promoting software innovation.

In addition to industry experts, John Dvorak, a prominent computer journalist, also feels new legislation is needed to expressly limit copyright protection for software. Dvorak writes:

> What we need is some new form of protection - not copyright or patent - specifically designed for software. It should protect ONLY the underlying source code from blatant rip-offs. That's what everyone in the business says should be protected. That's what everyone has always said should be protected.

Industry experts and commentators strongly support protection for source and object code. At the same time, they strongly oppose protection for non-literal software elements, in part because they fear it will produce more litigation, not innovation. Users also oppose copyright protection for non-literal software elements because it reduces the opportunity for standardized software and increases litigation.

B. *Expanding copyright protection for non-literal software elements contradicts users' desire for standardized software.*

The purpose of copyright protection is to provide an economic incentive for innovation in order to promote the public interest.

---

65. Two major software cases are currently pending: Lotus v. Borland and Apple v. Microsoft/Hewlett-Packard. John C. Dvorak, supra note 47, at 222. The legal and economic importance of these cases is huge. Apple is seeking $4.4 billion in damages from Microsoft, $3 billion in reduced unit sales and lower selling prices, and $1.4 billion in gross revenues Microsoft realized on the sale of Windows and related applications. Rory O'Connor, *Apple puts $4.4 billion price on suit,* SAN JOSE MERCURY NEWS, February 12, 1992, at Cl. On April 14, 1992, U.S. district Judge Vaughn Walker dismissed 46 of 69 disputed claims by Apple. The judge ruled that many elements of Microsoft's Windows, such as the use of "icons" and "windows," do not infringe on any of Apple's copyrights. In addition, the judge ruled that many aspects of the Macintosh operating system are not protected by copyright law. Rory O'Connor, *Judge favors Microsoft in "look and feel" dispute,* SAN JOSE MERCURY NEWS, April 15, 1992, at 1A.

66. Developers are unsure whether they should continue to develop, lobby for legislative change, or wait for the courts to resolve these issues. One computer lawyer said, "I've already had several [software developer] clients call to ask, 'What are we supposed to do?'" Neil Margolis, supra note 46, at 8.

67. David Reed explains the benefit of protection: "Copyright protection allows and encourages me to make the maximum commitment to my design work, knowing that the time-honored tradition of copyright will protect my creations from developers driven more by money than muse." David Reed, supra note 51, at 21.

68. Dvorak, supra note 47, at 222.
Computer users believe that standardized software will make computers easier to use. Users are in favor of standards because they prevent learning extra commands to run new computer software.\(^{69}\) One commentator has written that the substantially similar expression (i.e., use of well-designed user-interfaces in multiple programs) of ideas is the best thing to happen to computers since the machines moved to the desktop.\(^{70}\) Standardization allows users to use the same commands on different programs.

Increased copyright protection for computer software prevents standardization for two reasons. First, increased protection prevents subsequent developers from creating compatible products because the original developer will claim the new, improved product is an infringement. Second, the uncertainty over the appropriate extent of protection results in litigation and increased software costs which are passed on to the consumer.

1. Users' views receive inconsistent attention from the courts.

Users' desires regarding standardization are seldom heard in court because most suits are between software companies. Ned Saltzberg is a partner at Boston-based Warner & Stackpole who represented Stephenson Software, one of the defendants that Lotus defeated along with Paperback. Saltzberg believes:

Users need an advocate [in these cases] and so far, there hasn't been one. Maybe in the next suit, a user group or organization will file an amicus brief. It is users who have clamored for standards; it is users who will be inconvenienced to the extent that developers feel crushed between the competing interests of satisfying the customers and avoiding lawsuits.\(^{71}\)

a. Courts have held that common user commands may be unprotectible ideas - particularly if market factors are considered.

Some court decisions have recognized the standardization argument.\(^{72}\) In *Ashton-Tate Corp. v. Ross*,\(^{73}\) Ross claimed he deserved compensation for contributing elements of the user interface to

\(^{69}\) *Evolution of Software Copyright Lawsuits*, INFOWORLD, July 23, 1990, at 44.


\(^{71}\) *Evolution of Software Copyright Lawsuits*, supra note 69, at 44.

\(^{72}\) For an excellent summary of judicial recognition of the standardization issue, see Lewis, *supra* note 31, at 701-10.

\(^{73}\) 728 F. Supp. 597 (N.D. Cal. 1989), aff'd, 916 F.2d 516 (9th Cir. 1990).
Ashton-Tate’s spreadsheet program. However, the Northern District of California granted summary judgment to Ashton-Tate and held that user interface commands were unprotectible ideas under Section 102(b) of the Copyright Act. The court based its holding on the fact that the commands were common, were already available on other software programs, and lacked originality. Ashton-Tate implicitly recognized the public benefit in free access to recognizable, common user commands that are available on other computer programs.

Similarly, in Telemarketing Resources v. Symantec Corp., the Northern District of California recognized that certain user interface elements are “fundamental to a host of computer programs.” The Symantec court went further than Ashton-Tate, holding that copyright protection is inappropriate for commonplace elements of user interfaces. After Ashton-Tate and Symantec, commonplace user interface elements may not receive copyright protection.

In Plains Cotton Coop. Ass’n v. Goodpasture Computer Services, Inc., the court recognized that market factors play a significant role in determining the sequence and organization of software, and held that these patterns may constitute “ideas” in a computer context. The court realized that if users prefer a standardized format, granting copyright protection to that format would prevent other developers from designing competing programs. In spite of the holdings in Ashton-Tate, Symantec, and Plains Cotton, the court curtly dismissed the standardization argument in Lotus v. Paperback.

b. Lotus v. Paperback rejected the standardization argument.

Paperback Software admitted that VP-Planner copied the menu structure of Lotus 1-2-3. However, Paperback argued that to be a commercial success, VP-Planner would have to be compatible with Lotus 1-2-3, one of the largest installed application programs in the United States. In addition, the high costs of training requires a competitive product to be compatible with Lotus 1-2-3 to

74. Id. at 602.
75. Id.
76. 1990 Copyright L. Dec. (CCH) ¶26,514 at 23,084 (N.D. Cal. Sept. 6, 1989).
77. Id. at 23,087.
78. Id.
79. 807 F.2d 1256, 1262 (5th Cir. 1987), cert. denied, 484 U.S. 821 (1987).
80. Id.
81. Lotus v. Paperback, 740 F. Supp. at 69. There are seven million users of Lotus'
be marketable.  
Judge Keeton rejected Paperback Software's argument on two grounds. First, it ignored the commercial success of Excel, an innovative spreadsheet program with a menu structure that is not compatible with Lotus 1-2-3.  
Second, there is no statutory provision or precedent declaring that standardization is in the public's best interest.  
Judge Keeton ignored the standardization argument in Lotus v. Paperback. However, Judge Keeton could have considered the standardization argument in two ways. First, under the Ashton-Tate/Symantec rationale, the Lotus 1-2-3 user interface commands, which are commonplace, are unprotectible ideas. Competing products that use these elements would not be considered infringements. Second, defining the idea as a "marketable spreadsheet which adheres to the prevailing standardized user interface," instead of only a spreadsheet, would allow similar or identical user interfaces to be used by competing software products without the automatic threat of copyright infringement. By ignoring standardization, Lotus v. Paperback failed to adequately consider the public interest in copyright law.

C. The uncertainty in the current case law creates additional litigation, which prevents standardization and harms users through increased prices.

The courts continue to expand copyright protection for user interface elements. Allowing protection for common user interface elements is contrary to the views of software industry experts, which have remained innovative over the last decade in large part through the open exchange of information, and software users,

82. "Companies spend hundreds of millions of dollars every year training employees how to use standard-setting software such as 1-2-3. . . . They are unlikely to buy another program, no matter how improved, that requires major retraining." David E. Sanger, A Divisive Lotus clone 'War', THE N.Y. TIMES, Feb. 5, 1987, at A1.
83. Lotus v. Paperback, 740 F. Supp. 37 at 69, 78. Excel is not command-compatible with Lotus 1-2-3. However, Excel is data-compatible. Data from Lotus 1-2-3 can be translated into Microsoft Excel format with a single command. This data-compatibility is an option that Paperback Software could have chosen for VP-Planner. The court in Lotus v. Paperback states that "copying the menu structure was not the only way to achieve . . . compatibility." Id. at 69.
84. Id. at 79.
85. Lewis, supra note 31, at 705.
86. The open exchange of techniques and procedures helped create innovative software which accelerated the growth of the computer industry. Dan Bricklin did not try to seclude
who prefer standardization.

A major problem with the current protection is the case-by-case nature of the protection. The Ninth Circuit in Johnson Controls Inc. v. Phoenix Control Systems, Inc.\textsuperscript{87} stated that a case-by-case approach is necessary because:

whether the non-literal components of a program, including the structure, sequence, and organization and user interface, are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself.\textsuperscript{88}

Case-by-case decisions require more litigation to clarify the boundaries of the appropriate scope of copyright protection for non-literal software elements. This explains the current plethora of software litigation. As a result, financial resources which could be spent on innovative new products, product upgrades, and improved product service are wasted on litigation over legal issues that could be resolved by the legislature. Users pay the price by effectively subsidizing these legal battles in the form of high-priced software upgrades and inadequate product support.\textsuperscript{89} Another consequence of the case-by-case method is that the uncertainty causes many software developers to avoid using anything remotely similar to user interface elements in other programs, which makes standardization difficult, if not impossible.

Legislative change is a better solution than the inherent inconsistencies of decisions on a case-by-case basis. In the case of user interfaces, legislative protection would clarify what elements of user interfaces are protectible and what specific types of software elements are infringing.

VisiCalc, the first commercial spreadsheet, developed for the Apple II, from other programmers. Instead, he challenged them to try and make something better. Mitch Kapor took that challenge, and created Lotus 1-2-3. The Windows environment on the Macintosh and IBM personal computers started at Xerox Laboratories. Xerox initially chose not to sue Apple when Apple employees used the concept on both the Lisa and Macintosh computers. Open exchange of information helped create improved spreadsheets and windows environments. Currently, however, innovation is taking a back seat to trade secrets, nondisclosure agreements, and litigation. For interesting reading on how innovation occurred at Apple on the Macintosh project, see \textit{John Sculley, Odyssey} 154-182 (1987).

\textsuperscript{87} 886 F.2d 1173, 1175 (9th Cir. 1989).
\textsuperscript{88} \textit{Id.}
\textsuperscript{89} See MacGregor, supra note 49, at F5.
III. **Proposed Solution: Software Industry Must Lobby for Legislative Change to the Federal Copyright Act**

One software industry expert had the following comment regarding the ability of the courts to resolve the issue of protection of non-literal elements of software:

The arguments and session made me very nervous because the arguments against strong protection were so compelling based on my knowledge of the field, but they may not be anywhere near as obvious to nonpractitioners - and the courts are generally nonpractitioners.90

This indicates a reluctance to leave the level of protection in the court's hands. This reluctance should create a willingness to lobby for legislative change.

There are two types of change needed to meet the software industry's goals of protecting innovation and protecting original software designs. First, the software industry should ask Congress to expressly protect only source and object code. Second, the software industry must ask Congress to disallow copyright protection to non-literal elements of computer software except in limited circumstances. Specifically, copyright protection should be allowed to prevent the intentional cloning of another software product.

A. *The proposed amendment to the Federal Copyright Act.*

The following federal legislation is offered as a potential solution to the diverging views of the computer software industry and the federal courts.

The Computer Software Protection Act of 1992

1. **Definitions**—the following definitions apply to this statute:
   - The term "Federal Copyright Act" shall refer to Title 17 U.S.C. 101 et. seq.
   - "Literal elements" of computer software include source code and object code.
   - "Non-literal elements" of computer software shall include, but not be limited to, the sequence, structure, organization and "look and feel" of computer software.
   - "Object code" is program code that is understandable by the computer, and shall include commercial versions of source code.

90. Samuelson & Glushko, *supra* note 60, at 487.
that have been run through a compiler program so that the computer can read the program.

"Public domain software" includes any computer program that the author chooses to distribute for free through computer groups or modem services.

"Source code" is program code that is understandable by human programmers.

2. Literal elements of computer software shall be entitled to copyright protection under the Federal Copyright Act, provided it meets all the requirements of the Federal Copyright Act. All rights and liabilities that the author of the source code or object code would be entitled to under the Copyright Act shall remain available to the author.

3. Non-literal elements of computer software shall be entitled to no copyright protection under the Federal Copyright Act unless the following conditions are met:
   a. The alleged infringer has knowledge of the protected software, either actual or constructive; and
   b. The alleged infringing product is a clone, not an industry advancement. Factors to consider when deciding whether a product is a clone or industry advancement include: (1) relative prices of the two software packages; (2) amount of innovation beyond the infringement that is found in the product; and (3) industry experts' opinions on the classification of the product.

4. If a non-literal element meets the requirements in Sections 3(a) and 3(b), it is entitled to protection under the Federal Copyright Act, provided all the requirements of the Copyright Act are satisfied.

B. Pros and cons of the proposed legislation.

This legislation explicitly protects source code and object code, which is what industry experts and commentators would like protected. This legislation also protects non-literal elements of computer software, but only if the author of the infringing product had notice of the protected program, and the infringing product is not an industry advancement. This limited protection of non-literal elements prevents "rip-off" products that offer no advance for the industry.

The proposed legislation has two major advantages.91 First,
industry experts’ opinions are used to decide the status of the programs. This reduces the need for federal judges to have technical expertise in computer software. If industry experts feel the new software is an advancement, a court would be less likely to find the new software an infringement. However, if industry experts feel the product is a clone that does not advance the industry, a court would be more likely to find the new software an infringement. The new legislation insures that the software industry, the people protected by the copyright law, have a voice in the enforcement of the law. The proposed legislation allows courts to work with industry experts to insure that industry views are considered when determining if a software product is an infringement.

Second, explicit protection of literal elements of computer software is spelled out. For these items, protection under the Copyright Act remains unchanged. Non-literal elements of computer software, on the other hand, are only protected in limited circumstances. This reduces the uncertainty of the current state of protection. In addition, innovation is not discouraged because industry advancements will defeat any infringement claims.

IV. Conclusion

The Federal Copyright Act is unclear how much copyright protection should be given to non-literal aspects of computer software. Recent federal court decisions have expanded protection of non-literal software elements. However, computer software industry experts and commentators feel copyright protection of non-literal elements of computer software is not needed for innovation.

One solution to the conflicting views of the courts and industry experts is the proposed legislation offered above. The legislation provides express copyright protection for source code and object code. The legislation provides no copyright protection for non-literal elements of software unless: (1) the infringing software developer had knowledge of the protected software; and (2) the infringing software is not an industry advancement.

The proposed legislation will prevent unnecessary litigation. The public interest is provided for by allowing subsequent developers to use common interface elements as long as they improve the state of the art. Innovation is provided for by allowing strong copyright protection for software companies whose programs are un-

---

clean rooms and the legality of reverse engineering of software may create other problems with this proposed legislation. These items are beyond the scope of this Comment.
fairly copied by subsequent developers. Under the proposed legislation, Lotus would be successful in an infringement suit against Paperback Software because VP-Planner is a "clone" that does not advance the industry. However, Lotus would be unsuccessful in its current infringement suit against Borland because Quattro is an industry advancement and not an infringing product.