

NLWJC - Kagan

Counsel - Box 031 - Folder 006

Lotus v. Borland [2]



U. S. Department of Justice
Office of the Deputy Attorney General

Associate Deputy Attorney General

Washington, D.C. 20530

December 1, 1995

Ms. Elena Kagan
Associate Counsel to the President
OEOB 125
Washington, D.C. 20500

Dear Elena:

Attached is a copy of the Antitrust Division's draft brief in the Lotus v. Borland matter. This is the most current draft, and is the draft from which the Solicitor General is working. But it does not reflect the work of his office.

His office is confident it can meet many of the PTO concerns, as we discussed.

Please let me know if I can be of further assistance.

Sincerely,

David W. Ogden

AFT November 17, 1995 2:43 pm

No. 94-2003

IN THE SUPREME COURT OF THE UNITED STATES

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION, PETITIONER

v.

BORLAND INTERNATIONAL, INC.

ON WRIT OF CERTIORARI
TO THE UNITED STATES COURT OF APPEALS
FOR THE FIRST CIRCUIT

BRIEF FOR THE UNITED STATES AS AMICUS CURIAE
SUPPORTING RESPONDENT

DREW S. DAYS, III
Solicitor General

ANNE K. BINGAMAN
Assistant Attorney General

LAWRENCE G. WALLACE
Deputy Solicitor General

JOEL I. KLEIN
Deputy Assistant Attorney General

BETH S. BRINKMANN
Assistant to the Solicitor General

CATHERINE G. O'SULLIVAN
DAVID SEIDMAN
Attorneys

Department of Justice
Washington, D.C. 20530
(202) 514-2217

DRAFT November 17, 1995 2:43 pm

QUESTION PRESENTED

Whether a computer menu command hierarchy, which defines the meaning of keystrokes when used in particular sequences, is an "idea, procedure, process, system, method of operation, concept, principle, or discovery" within the meaning of section 102(b) of the Copyright Act, 17 U.S.C. §102(b).

(I)

DRAFT November 17, 1995 2:43 pm

IN THE SUPREME COURT OF THE UNITED STATES

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION, PETITIONER

v.

BORLAND INTERNATIONAL, INC.

ON WRIT OF CERTIORARI
TO THE UNITED STATES COURT OF APPEALS
FOR THE FIRST CIRCUIT

BRIEF FOR THE UNITED STATES AS AMICUS CURIAE
SUPPORTING RESPONDENT

INTEREST OF THE UNITED STATES

This case involves the copyright status of a menu command hierarchy used in a copyrighted computer program. Copyright law reflects a balance Congress struck between "competing claims upon the public interest." Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975). It "seeks to establish a delicate equilibrium. On the one hand, it affords protection to authors as an incentive to create, and, on the other, it must appropriately limit the extent of that protection so as to avoid the effects of monopolistic stagnation." Computer Assoc. Int'l. Inc. v. Altai, Inc., 982 F.2d 693, 696 (2d Cir. 1992).

DRAFT November 17, 1995 2:43 pm

The United States has a substantial interest in "the preservation of the balance between competition and protection reflected in the . . . copyright laws." Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971). It has primary responsibility for enforcing the antitrust laws, which establish a national policy favoring economic competition as a means to advance the public interest. It also has responsibilities for administering and enforcing the copyright laws. Thus, the United States is properly concerned that interpretation of the copyright laws not depart from the congressionally struck balance.

STATEMENT

1. Lotus Development Corp. ("Lotus") markets a copyrighted computer program known as 1-2-3.^{1/} 1-2-3 is an electronic spreadsheet, a computer program that performs operations, at a user's direction, on data organized and displayed in rows and columns like those of a paper spreadsheet. 1-2-3 marked a significant advance over prior spreadsheets. Pet. App. 230a-231a.

The spreadsheet user tells the program what operations to perform by giving it commands. Lotus 1-2-3 recognizes more than 400 commands. Pet. App. 129a. Following the lead of the first

^{1/}Several versions of 1-2-3 exist. We ignore the distinctions here.

DRAFT November 17, 1995 2:43 pm

commercially successful electronic spreadsheet, Lotus organized
1-2-3's commands by using

a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command ... so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy. (Pet. App. 129a.)

1-2-3 displays one menu of commands at a time, and the user may directly select commands on a menu only while it is displayed.

A user may select commands, or operations, by causing the program to display menus (in 1-2-3 and many other spreadsheets by striking the "/" key, Pet. App. 232a) and then selecting menu items by striking the key corresponding to the first letter of the desired command (or highlighting the word referring to the command and pressing the enter key). The user may be guided by the program's displays but also may, like a touch typist, strike keys without relying on displays. The command, if any, to which a given letter corresponds depends on the menu displayed and therefore on the sequence of commands previously chosen; that is, it depends on the structure of the command hierarchy. For example, in the sequence "/FR," "F" stands for the File command (which invokes the File submenu) and "R" for the Retrieve command (which prompts for a file name and then retrieves the file). In the sequence "/RF," the "R" stands for the Range command (which

DRAFT November 17, 1995 2:43 pm

invokes the Range submenu) and "F" for the Format command (which invokes the Format submenu). Richard H. Stern, Copyright in Computer Programming Languages, 17 Rutgers Computer & Tech. L.J. 321, 330 n.19 (1991) ("Stern"). See also Pet. App. 110a-111a ("C" may invoke Currency or Copy, depending on context). Sequences that do not correspond to paths through the hierarchy, such as "/FF" and "/RR", have no meaning. Stern, supra, at 330 n.20.

Instead of entering keystroke sequences from the keyboard as the program responds to them, "the user may store a sequence of command terms as a 'macroinstruction,' commonly called a 'macro,' and then, with one command stroke that invokes the macro, cause the programmed computer to execute the entire sequence of commands." Pet. App. 228a-229a. Macros may also contain various nonmenu commands, such as one that "performs an 'if-then' logical function," id. at 229a, which give the 1-2-3 macro language a "sophisticated programming capability." Pet. App. 110a. Macros written in this language are computer programs that can control the operation of 1-2-3 and thus, indirectly, the computer.²

At relevant times, Borland International, Inc. ("Borland") marketed a spreadsheet program known as Quattro.³ Quattro has a

²For example, the following sequence, stored as a macro, will sort the entire first column of the spreadsheet in ascending order: /dsd{home} .{end}{down}~p~a~

³Several versions of Quattro exist. We ignore the distinctions here. Subsequent to the court of appeals' decision, Borland sold Quattro to another company.

DRAFT November 17, 1995 2:43 pm

"native" command hierarchy that differs significantly from the 1-2-3 command hierarchy. See Pet. App. 108a. In addition, however, Quattro also permitted users to select an alternative "emulation" mode that incorporated the 1-2-3 command hierarchy and thus presented users with menus containing the same commands and in the same order as the 1-2-3 menus (along with some commands not found in the 1-2-3 menus). Pet. App. 82a-84a. Although Quattro's visual presentation of these menus did not mimic that of 1-2-3, the emulation mode permitted the user to select operations by using the keystroke sequences used in 1-2-3. It also allowed Quattro to execute macros written in the 1-2-3 macro language. Borland later removed Quattro's emulation mode, replacing it with a "key reader," which allows Quattro to execute macros written in the 1-2-3 macro language without displaying menus containing those commands.⁴

2. Lotus filed suit in July 1990, alleging, in substance, that Borland copied Quattro's emulation mode from the 1-2-3 "user interface" and thereby infringed the 1-2-3 copyright. Lotus did not allege that Borland had copied any of the "statements or instructions" constituting the program code of 1-2-3.⁵

⁴Certain macro commands cause the macro to pause while the user performs keystrokes. The key reader permits Quattro to interpret these keystrokes as if the emulation-mode menus were displayed.

⁵A 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.

DRAFT November 17, 1995 2:43 pm

After the district court held, on summary judgment, that there was no genuine dispute that Quattro infringed but that issues of fact as to the scope of the infringement remained, Pet. App. 143a, Borland removed the 1-2-3 emulation from Quattro, and Lotus filed an amended complaint alleging that the key reader infringed. Following a bench trial on the remaining liability issues, the district court issued separate opinions concerning issues relating to the 1-2-3 emulation mode (Pet. App. 71a-105a) and issues relating to the key reader (Pet. App. 29a-68a).

3. The district court held that the 1-2-3 copyright protected the menu command hierarchy, including both the command words and the menu structure. Acknowledging that ideas are not copyrightable, and that the idea of the 1-2-3 user interface may be articulated at different levels of abstraction, the court chose the following description of that idea:

Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command . . . so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy. (Pet. App. 129a.)

The court then ruled that the 1-2-3 interface had "identifiable elements of expression not essential to every expression of that idea," because a "very satisfactory spreadsheet menu tree can be constructed using different commands

DRAFT November 17, 1995 2:43 pm

and a different command structure from those of Lotus 1-2-3."

Pet. App. 131a. Moreover, "it cannot be genuinely disputed that a large part of the structure and arrangement of the menu commands is not driven entirely by functional considerations."

Pet. App. 133a. Finally, the court concluded that the expressive elements of the menu commands, menu command hierarchy, macro language, and keystroke sequences were a substantial part of 1-2-3 (Pet. App. 134a); that Borland had copied these elements of the interface (id. at 115a); and that this copying rendered Quattro "substantially similar to 1-2-3" (id. at 138a). Thus, Borland had infringed the 1-2-3 copyright.

The court also held that Borland's key reader infringed. It concluded that "the Lotus menu structure and organization (including the first letter of the commands, used to mark the structure) are part of the protectable expression found in the Lotus 1-2-3 program." Pet. App. 44a. The court then found that the Quattro file containing the information necessary to the key reader included a "virtually identical copy of the Lotus menu structure," id. at 47a, although the tree structure is "represented in a different form." Id. at 35a. The court explained that Borland had "translated (nearly verbatim) the menu structure into a different language for representing menu structures," id. at 47a, and it viewed translation in general as copying. Ibid., (citing 17 U.S.C. 101 ("`derivative work' includes translation")). Because the court had previously found

DRAFT November 17, 1995 12:43 pm

the menu structure to be protected by copyright, ~~the~~ conclusion of infringement directly followed.

Implementing these various rulings, the district court entered an order permanently enjoining Borland from distributing versions of Quattro containing "in any portion, component or module thereof, a copy of the Lotus 1-2-3 menu commands and/or menu structure, in any form." Pet. App. 69a-70a.⁶ Borland appealed from that order.

4. The court of appeals reversed unanimously. The sole issue before it was "whether a computer menu command hierarchy is copyrightable subject matter." Pet. App. 3a. (Lotus did not "contend on appeal that the district court erred in finding that Borland had not copied other elements of Lotus 1-2-3, such as its screen displays." *Id.* at 10a.⁷) The court held that the menu command hierarchy was not copyrightable and therefore that

⁶The order further provided that "Borland may at any time . . ." apply to this court for modification of this clause to permit marketing of a product that does not infringe or if a remedy involving payment of royalties were more appropriate. Pet. App. 70a.

⁷Thus, the court emphasized, it had no occasion to opine on the copyrightability of Lotus' screen displays (Pet. App. 16a n.10), its "long prompts" explaining choices available to the user (*id.* at 16a n. 9), or its program code (*id.* at 16a-17a n. 11). After the court of appeals issued its mandate, the district court vacated the injunction. Order Vacating Permanent Injunction, May 25, 1995. Pursuant to local rules, the case was transferred to a different district judge, and Borland moved for entry of judgment, contending that no issues remained to be resolved. The district court declined to enter judgment, Order, June 30, 1995, and Borland has appealed and filed a petition for mandamus. These matters are pending before the court of appeals as docket numbers 95-17933 and 95-1885.

DRAFT November 17, 1995 2:43 pm

"Borland did not infringe Lotus's copyright by copying it." Id. 22a.⁸

The court of appeals first observed, "[c]omputer programs receive copyright protection 'literary works,'" Id. 11a n.5, and Borland did not dispute that "Lotus has a valid copyright in Lotus 1-2-3 as a whole." Pet. App. 11a.⁹ But the district court did not rest its injunction primarily on a finding that Borland had copied the "words, numbers, or other verbal or numerical symbols or indicia," 17 U.S.C. 101, constituting the literal expression of Lotus's computer program.¹⁰ Nor, in the

⁸Because the court so held, it did not consider Borland's affirmative defenses, such as fair use. Pet. App. 22a.

⁹The Copyright Act protects literary works, 17 U.S.C. 102(a)(1), defined to include "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, . . . , tapes, disks, or cards, in which they are embodied," id. 101. The House Report explained that the term includes "computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves." H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5667. In 1980, Congress added to the Act a definition of "computer program" that makes clear that computer programs fall within the definition of literary works: "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.

¹⁰Borland's emulation mode, but not its key reader, included full words like "Copy" and "Print" used by Lotus to identify commands on screen. But, as Lotus explains, "the words do not, in themselves, perform any mechanical function [but merely] provide information to assist users in selecting the appropriate instructions to cause the program to perform certain tasks." Br. 6. Lotus's copyrighted program code undoubtedly contains instructions that cause these words to appear on a computer screen as part of the user interface, Br. 7, but there is no allegation that Borland copied these instructions. In any event,

DRAFT November 17, 1995 2:43 pm

court of appeals' view, did the appeal involve copying of the "nonliteral" elements of the program that are eligible for protection.^{11/} Id. at 14a Rather, the court described the issue as "literal copying of the Lotus menu command hierarchy." Id.

The court of appeals declined to apply the "abstraction-filtration-comparison" test set forth by the Second Circuit in Altai. Id.^{12/} In the court's view, the "initial inquiry should not be whether individual components of a menu command hierarchy are expressive, but whether the menu command hierarchy as a whole can be copyrighted." Id. at 15a.

The court concluded that Lotus' menu command hierarchy is uncopyrightable because it is a "method of operation" foreclosed from copyright protection by section 102(b) of the Copyright Act,

because the district court found not only the emulation mode, but also the key reader, to infringe, it is clear that its finding of infringement did not turn on literal copying of these words. The copyrightability of Lotus's screen displays was not before the court of appeals. Pet. App. 16a n.10.

^{11/}As in the case of other literary works, copyright protection of a computer program is generally understood to extend beyond the literal elements of source code and object code to "'nonliteral' elements, such as the program architecture, 'structure, sequence and organization', operational modules, and computer-user interface." Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.2d 1335, 1341 (5th Cir. 1994) ("Most courts confronted with the issue" have so held, citing, inter alia, Computer Assocs. Int'l., Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Gates Rubber Co. v. Bando Chemical Indus., Ltd., 9 F.3d 823 (10th Cir. 1993)). As we show below, the command hierarchy the district court protected, as revealed by its finding that the key reader infringed, is a non-literal element of 1-2-3.

^{12/}As discussed below, see n. , we believe the court of appeals' analysis is consistent with Altai.

DRAFT November 17, 1995 2:43 pm

17 U.S.C. § 102(b). Pet. App. 15a-22a.^{13/} Having reached that conclusion, the court did not consider whether the command hierarchy was also a "system, process, or procedure" within the meaning of the statute, as Borland had argued. Id. at 15a.

The court explained that it understood "method of operation" to "refer to the means by which a person operates something, whether it be a car, a food processor, or a computer." Id. The menu command hierarchy was a method of operation because it "provides the means by which users control and operate Lotus 1-2-3." Id. Noting the district court's holding that the hierarchy "constituted an 'expression' of the 'idea' of operating a computer program with commands arranged hierarchically into menus and submenus," Pet. App. 17a, the court concluded that any such "expression is not copyrightable because it is part of Lotus 1-2-3's 'method of operation.'" For, "[i]f specific words are essential to operating something, then they are part of a 'method of operation' and, as such, are unprotectable." Id. Accordingly, in the court's view, although "copyright assures authors the right to their original expression," Feist Publications, Inc. v. Rural Telephone Co., 499 U.S. 340, 349-350 (1991), that assurance is subject to the qualification that original expression that "falls within one of the [section

^{13/}Section 102(b) provides: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."

DRAFT November 17, 1995 2:43 pm

102(b)] categories [is] foreclosed from copyright protection."
Pet. App. 21a.

Judge Boudin concurred, observing that because a "computer program is a means for causing something to happen," copyright protection for computer programs "can have some of the consequences of patent protection." Pet. App. 23a. Accordingly, "[a]pplying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit." Id. 24a. In solving that puzzle, Judge Boudin concluded that this case "is an unattractive one for copyright protection of the menu," id. 26a, in essence because users who invested in learning the command hierarchy and in writing macros might not switch to a later-developed but better spreadsheet if doing so required that they abandon those investments. Id. 26a-27a. Therefore, the question was "not whether Borland should prevail, but on what basis." Id. 27a. In Judge Boudin's view, the court's focus on "method of operation" as an answer to that question was "defensible," id., even though section 102(b) "if taken literally might easily seem to exclude most computer programs from [copyright] protection." Id. at 24a.

SUMMARY OF ARGUMENT

The court of appeals reached the right result, but for the wrong reason. We agree with petitioner that section 102(b) of the Copyright Act, 17 U.S.C. §102(b), codifies the idea/expression dichotomy established in Baker v. Selden, 101

DRAFT November 17, 1995 2:43 pm

U.S. 99 (1880). We further agree that the court of appeals seriously distorted the meaning of the statute by ignoring that fundamental distinction in favor of a narrow focus on the illustrative phrase "method of operation" and by reading that phrase to reach both idea and expression. Although that misreading did not lead the court to an erroneous result, we, like petitioner, fear that its interpretation could, in another case, effectively nullify Congress' decision to classify computer programs, all of which serve to operate a computer, as literary works eligible for protection under the Copyright Act.

Despite our disagreement with the court of appeals' reasoning, however, we believe that its result was correct because Lotus' menu command hierarchy, as protected by the district court, is an abstract idea unprotectable under section 102(b). That command hierarchy, more abstract than the visual menus that provide information to users, is a system of rules that defines permissible sequences of symbols, expressed as keystrokes or otherwise, and assigns meaning to those sequences. It is distinguishable from the particular form in which Lotus presents information and choices to the user; from the program code that causes the computer to present that information or process sequences a user enters; and from the manner in which those rules are expressed in 1-2-3. The hierarchy itself does not instruct the computer to carry out any function; rather, it is the structure of a language that allows the user and 1-2-3 to

DRAFT November 17, 1995 2:43 pm

communicate. It may facilitate expression, but it is not itself expression.

The district court's order, which the court of appeals properly reversed, broadly enjoined Borland from distributing any version of its spreadsheet capable of interpreting and carrying out user commands expressed in the Lotus 1-2-3 language. Whether or not Borland's emulation mode involved other "copying," the district court's fundamental premise was that any manner of representation of the Lotus command hierarchy that would permit a computer program to understand user commands so expressed necessarily copied protected expression.

The district court's ruling thus signalled a departure from the widely held understanding that the formal structure of a language is not itself copyrightable. Acceptance of its reasoning could afford the originator of a computer language broad-ranging power to prevent or regulate others' creation or use of original programs designed to interact with the copyright owner's program, to translate among languages, or -- as in this case -- to interact with computer programs written by users of the copyright owner's program in such a language. Such a result, erecting substantial barriers to expression and competition alike, cannot be reconciled with the policies embodied in the Copyright Act.

ARGUMENT

- I. CONTRARY TO THE COURT OF APPEALS' INTERPRETATION, SECTION 102(b) OF THE COPYRIGHT ACT REFLECTS THE LONG-ESTABLISHED

DRAFT November 17, 1995 2:43 pm

DISTINCTION BETWEEN COPYRIGHTABLE EXPRESSION AND
UNCOPYRIGHTABLE ABSTRACT IDEAS

As this Court has explained, "[t]he primary objective of copyright is . . . '[t]o promote the Progress of Science and useful Arts.' Art. I, §8, cl.8. . . . To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas . . . conveyed by a work. . . . This principle, known as the idea/expression . . . dichotomy, applies to all works of authorship." Feist, 499 U.S. at 349-350. Under this fundamental principle, ideas -- regardless of their originality, creativity, or importance and regardless of the effort involved in generating them -- may not be copyrighted. An original expression of an idea, on the other hand, ordinarily may be protected, no matter how trivial or uninteresting the idea expressed.

The seminal case applying the idea/expression dichotomy, if not its terminology, is Baker v. Selden, 101 U.S. 99 (1880). Charles Selden had copyrighted several books on a system of bookkeeping, based on the conventional double entry method, that presented operations for various time periods on one or two pages. The books explained the system and included forms illustrating it. Baker subsequently published "account books arranged on substantially the same system." 101 U.S. at 101. Selden's widow sued for copyright infringement. The circuit court found that Baker's books "are in large and material part identical with" Selden's books and enjoined their publication and

DRAFT November 17, 1995 2:43 pm

distribution. Selden v. Baker, No. 1620, C.C.S.D. Ohio, Jan. 21, 1875, reprinted in Baker v. Selden, 101 U.S. 99 (1980), Transcript of Record at 8, 9.¹⁴

This Court reversed, concluding that the copyright in Selden's book "did not confer upon him the exclusive right to make and use account-books, ruled and arranged as designated by him and described and illustrated in said book." 101 U.S. at 107. The Court distinguished "between the book, 'as such, and the art which it is intended to illustrate" and observed that property rights in the art were "the province of letters patent, not of copyright." Id. at 102. Copyright was no bar to use of the art, the "rules and methods" of that art, or "the ruled lines and headings of accounts [that] must necessarily be used as incident to it." Id. at 104. In other words, publication of the

¹⁴The circuit court found that "the device, method, and form of [Baker's] books for entering all the items of all monies received and disbursed, item by item, each item as to its proper fund, are, as to the five left-hand columns employed by him, identical with and an infringement of the said Selden system; and that the device, method, and form of [Baker's] said book for aggregating these items with previous balances to their respective funds, and so as to show the condition and balance to the debit or credit of each of these funds, are, as to the column of funds, the two columns of brought forwards, the two columns of 'current receipts and disbursements,' the two columns of 'totals,' and the two columns of 'balances,' so far as these columns respect the funds, identical with and an infringement of the books of the said Selden system." Selden v. Baker, supra, at 9.

This Court did not explicitly reject the lower court's findings, and we do not believe it did so implicitly in correctly noting that Baker's work "makes a different arrangement of the columns, and uses different headings." 101 U.S. at 100.

DRAFT November 17, 1995 2:43 pm

art dedicates it to the public despite the copyright in the work making it available. Ideas are unprotected by copyright, although expression may be protected.^{15/}

Congress wrote against the backdrop of this settled understanding of copyright law when it enacted Section 102(b) in 1976. As this Court has recognized, Congress did not intend to alter, but to embody, the fundamental idea/expression dichotomy: "Congress emphasized that § 102(b) did not change the law, but merely clarified it: 'Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate . . . that the basic dichotomy between expression and idea remains unchanged.'" Feist, 499 U.S. at 356 (quoting H.R. Rep. No. 94-1476 at 57; S. Rep. No. 94-1473, at 54). The courts of appeal have also recognized that Section 102(b) embodies this "axiom of common law." Altai, 982 F.2d at 702; see also, e.g., Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1234 (3d Cir. 1986) ("intended to express the idea-expression dichotomy"), cert. denied, 479 U.S. 1031

^{15/}The Court also explained that if the art cannot be practiced without using methods and diagrams in the copyrighted work, or similar to those in the copyrighted work, those "methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public," at least for purposes of "practical application" of the art. 101 U.S. at 103. This Court's subsequent description of Baker v. Selden in Mazer v. Stein, 347 U.S. 201, 217 (1954), omits this aspect of Baker, although the principle remains that if protection of expression necessarily protects the idea expressed, idea and expression merge, leaving the expression unprotected. See, e.g., Altai, 982 F.2d at 707-708.

DRAFT November 17, 1995 2:43 pm

(1987); M. Kramer Mfg. Co., Inc. v. Andrews, 783 F.2d 421, 434 (4th Cir. 1986) (codification of principle that copyright protection extends only to expression of idea, not the idea); Apple Computer, Inc. v. Microsoft Corp., 35 F.2d 1435, 1443 & n.11 (9th Cir. 1994) (codifies principle of Baker v. Selden), cert. denied, ___ U.S. ___ (1995?). Cf. Gates Rubber, 9 F.3d at 836-837 (codifies both idea-expression dichotomy and "process-expression" dichotomy).

In light of this long established principle distinguishing idea from expression, clear congressional intent to incorporate that distinction in section 102(b), and close to uniform judicial interpretation consistent with that principle, it is now untenable to contend that Section 102(b) radically altered settled copyright principles by establishing a list of categories of things (methods of operation, systems, processes, etc.) that are unprotected by copyright as to both idea and expression. The multiple categories of Section 102(b) are most reasonably read as multiple illustrations of the kind of abstractions, conventionally summarized as "idea," that Congress intended to exclude from copyright protection. Thus, courts generally attempt neither to define nor to distinguish these categories. See, e.g., Pet. App. 15a (hierarchy is method of operation, but may also be system, process, or procedure, not defined); Gates Rubber, 9 F.3d at 868 & n.13 (grouping ideas, concepts, and principles under rubric of ideas, without definition; grouping

DRAFT November 17, 1995 2:43 pm

procedures, processes, systems, and methods of operation under rubric of process, defining them all as "a method for achieving a particular result").¹⁶

Not only does an approach that interprets each term in section 102(b) literally fail to respect congressional intent to embody the idea/expression dichotomy in Section 102(b), but it threatens to nullify Congress' decision to treat computer programs as literary works under the Copyright Act. The court of appeals acknowledged, Pet. App. at 11a n.5, that Congress made clear its understanding of the copyrightability of computer programs in 1976, and it expressly amended the Act to include a definition of computer program in 1980. As Judge Boudin observed, although Congress indicated in 1976 that computer programs "might be subject to copyright protection," at the same

¹⁶We do not, of course, suggest ignoring such terms as "method of operation" or "process," for the Court "should hesitate . . . to treat statutory terms [as surplusage] in any setting." Ratzlaf v. United States, 114 S. Ct. 655, 659 (1994). But terms in a statutory scheme may properly be read as illustrative of others without offending this canon of construction, and the terms that follow "idea" in section 102(b) are best read as illustrative of that underlying concept. "[T]he meaning of statutory language, plain or not, depends on context. 'Words are not pebbles in alien juxtaposition; they have only a communal existence; and not only does the meaning of each interpenetrate the other, but all in their aggregate take their purport from the setting in which they are used.'" King v. St. Vincent's Hosp., 112 S. Ct. 570, 574 (1991) (quoting NLRB v. Federbush Co., 121 F.2d 954, 957 (2d Cir. 1941) (L. Hand, J.) (citations omitted). Thus, "in construing a statute[,] a court should adopt that sense of words which best harmonizes with context and promotes [the] policy and objective of [the] legislature." Id. at 574 n.10 (quoting United States v. Hartwell, 73 U.S. (6 Wall.) 385, 396 (1868)). The interpretation of section 102(b) advanced here does just that.

DRAFT November 17, 1995 2:43 pm

time "Congress adopted a string of exclusions that if taken literally might easily seem to exclude most computer programs from protection." Pet. App. 25a. A computer program is, after all, "the means by which a person operates . . . a computer," Pet. App. 15a, and therefore presumably a method of operation.¹⁷ In short, "[s]ince one should endeavor to construe statutes in a way that does not render them futile, and since we know that Congress did determine in 1980 to protect computer programs, the terms 'process,' 'system,' or 'method of operation' must not be understood literally." Jane C. Ginsburg, "Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software," 94 Colum. L. Rev. 2559, 2570 (1994) (footnotes omitted).

The court of appeals' rationale for its holding thus rests on a fundamental misconstruction of Section 102(b).— By failing to give effect to Congress' purpose to protect expression while leaving idea unprotected, it raised unjustified doubts about the copyright status of any work of authorship that could be

¹⁷The court of appeals explained that although the command hierarchy was an uncopyrightable method of operation, the program code constituting the literal Lotus 1-2-3 was not, because the precise program code was not necessary to create a program with the capabilities of Lotus 1-2-3, while the precise menu hierarchy was necessary to allow users to operate Borland's program in the same way they operated Lotus's program. Pet. App. 16a. We understand neither how the distinction follows from the court's concept of "method of operation" nor what the distinction has to do with whether Lotus 1-2-3 is a means by which a user operates a computer.

DRAFT November 17, 1995 2:43 pm

characterized as "procedure, process, system, [or] method of operation." This is not the balance Congress struck in the Act.

II. ~~THE~~ MENU COMMAND HIERARCHY IS UNPROTECTED BY COPYRIGHT BECAUSE IT IS IDEA, NOT EXPRESSION, WITHIN THE MEANING OF SECTION 102(b)

We recognize that "[d]rawing the line between idea and expression is a tricky business." Altai, 982 F.2d at 704. Nevertheless, we believe that the menu command hierarchy protected by the district court is idea, not expression.

1. Although Quattro's key reader did not display any 1-2-3 command and there was no allegation that Borland's program code resembled Lotus's, the district court determined that the key reader file contained a "copy" of the Lotus menu tree structure, albeit "represented in a different form and with first letters of menu command names in place of the full menu command names." Pet. App. 35a. Because the key reader could interpret macros written in the 1-2-3 macro language, "the program must use the Lotus 1-2-3 menu structure." Id. 39a. As the court explained (ibid.):

If a program did not have a representation of the 1-2-3 menu hierarchy somewhere within the program code (or in a file that is used by the code), then there is no way that the program could understand that "rfc" refers to a path through a menu tree to the specific executable operation that changes a cell or cells [sic] appearance to monetary units (i.e., a path through the range and format menus to the currency leaf).

In other words, the copying at issue here was Borland's incorporation of the elements necessary to allow Quattro to

DRAFT November 17, 1995 2:43 pm

"understand" certain commands a user entered, without regard to the form those elements took in Borland's program or any other similarity between literal or nonliteral elements of the 1-2-3 and Quattro computer programs. The court protected an abstract conceptual organization or structure, however expressed, that gave meaning to single letters of the alphabet according to the order in which the user selected them.

So understood, the command hierarchy the district court protected constitutes the formal and abstract structure of a language.¹⁹ The command symbols, no more than individual

¹⁹See, e.g., Webster's Third International Dictionary 1270 (defining "language" as, inter alia, "a systematic means of communicating ideas or feelings by the use of conventionalized signs, sounds, gestures or marks having understood meaning" and "an artificially constructed primarily formal system of signs and symbols (as symbolic logic) including rules for the formation of admissible expressions and their transformations"); Donald Spencer, Webster's New Word Dictionary of Computer Terms 323 (5th Ed. 1994) (defining "language" as "[s]et of rules, representations and conventions used to convey information. A way of passing instructions to the computer other than through direct input of number codes."); J.E. Sammet, "Programming Languages" 1228-1229 in Encyclopedia of Computer Science and Engineering (Anthony Ralston, ed., 1983) ("A programming language is a set of characters and rules for combining them, which has" four characteristics that distinguish "programming languages" from other languages); J.A.N. Lee, "Programming Linguistics" 1232-1233 in [Ralston, again] ("Languages for communication between any two systems, be they human or mechanical, can be described by three intertwining concepts: syntax, semantics, and pragmatics," but because computer languages are artificial languages, "there exists no difference between the semantics and the pragmatics.")

DRAFT November 17, 1995 2:43 pm

letters, form the language's vocabulary,^{19/} and the hierarchy defines its syntax and semantics. See Stern, supra, at 327-330.

Such a formal structure defining a language is not a computer program, for the rules of the language do not directly (or indirectly, through translation into machine language) instruct the computer to perform any operation or "bring about a certain result." 17 U.S.C. 101. Rather, statements written in the language according to those rules -- i.e., macros -- constitute such instructions.

Lotus embodied the structure and rules of the language in, among other things, a set of menus 1-2-3 displays to the user. It may be argued that Lotus did not at that point intend to create a language, but the contemplated relationship between computer program and language is irrelevant to the nature of that which Lotus created. The author of a book may begin and end by writing the words of his text, with no conscious reliance on separately formulated ideas. But the book may nonetheless embody unprotected ideas. Lotus may have intended only to provide some menus to guide users in using 1-2-3. But in the process it created a language, which may be expressed in other ways.

^{19/}The first letters of commands are not necessary to the writing of macros, although they are necessary to the interpretation of macros written using those letters. In the hierarchy, meaning is assigned to position in the structure, so that symbols denoting uses of the computer's cursor arrow keys and the enter key can be used instead of letters. But this simply amounts to a translation of symbols into other symbols, which should not alter the analysis.

DRAFT November 17, 1995 2:43 pm

The formal structure of the language amounts to a set of rules governing both the writing of statements in that language and, as the district court made clear, the interpretation of those statements once they are written.²⁰ Any particular computer program that will interpret programs (macros) written in the language defined by the command hierarchy must contain a programmer's expression of a representation of that abstract set of rules. This is necessary whether that computer program be a spreadsheet, a separate program that translates Lotus macros into the language defined by the different command hierarchy of a different spreadsheet program, or a program that simply annotates the text of a macro with an English language rendition of the

²⁰We recognize that the command hierarchy does not define the entire Lotus macro language. That the hierarchy governs only a subset of the 1-2-3 macro language means that the language it specifies is limited, incapable of the sophisticated instructions possible in the larger macro language. Simple, unsophisticated computer programs, such as macros based entirely on a subset of the macro language, are no less computer programs than are more complicated and sophisticated sets of instructions to the computer, and the language in which they are written is no less a language.

In its Paperback decision, the district court declined to analyze the 1-2-3 user interface as a language. Pet. App. 242a-244a. That case involved allegations that Paperback had copied from the 1-2-3 user interface much more than the minimal symbolic commands and hierarchy, however, and the court apparently understood Paperback to use the term "language" much more broadly than we do here. Thus, the court noted that "language" has many meanings, and that "language" and "set[s] of statements or instructions" are not necessarily mutually exclusive categories. Pet. App. 244a. Because the latter clearly are copyrightable in some circumstances, the mere possibility of describing a communication as "language" cannot alone be determinative, the court concluded. Our argument turns on the nature of what the court protected, not on use of the term "language."

DRAFT November 17, 1995 2:43 pm

cryptic macro notation, so as to make it more understandable to humans and facilitate subsequent modifications. See Pet. App. 39a; Melville B. Nimmer and David Nimmer, Nimmer on Copyright § 13.03[F], at 13-144.4 n.336.10 (district court's "ultimate holding would render infringing any conceivable macro translation device") (1995) ("Nimmer").^{21/}

2. The critical question under section 102(b), which the court of appeals declined to address, is whether this abstract set of rules is idea or expression.^{22/}

The district court, which first addressed the idea/expression dichotomy prior to Borland's introduction of the key reader, effectively determined its result by starting with the conclusion that "the 'idea' or 'system' of the 1-2-3

^{21/}The district court quite properly did not hold that macro translation schemes, such as "one-time translation," that were not before it would infringe Lotus's copyright. Pet. App. 39a. But, as we have noted, it made clear that any such scheme must incorporate a representation of the command hierarchy. Thus, even on-time translation requires substantial copying of what the district court found to be protected.

^{22/}Petitioner observes that "a consensus has emerged" (Pet. Br. 37) around the "abstraction-filtration-comparison" test articulated by the Second Circuit in Altai. The court of appeals below believed the Altai analysis unhelpful in this context. Pet. App. 14a. The court's analysis, however, appears to be consistent with Altai. See Mark A. Lemley, Convergence in the Law of Software Copyright?, 10 High Tech. L.J. 1, 22 (1995). Professor Lemley explains that "[a] proper application of the Alta approach in this context would identify the menu command hierarchy as the level of abstraction at which copying was alleged." Id. The case came to the court of appeals with that level of abstraction already identified. The next step in the court's analysis corresponds to filtration; the court filtered out unprotectable elements in light of its understanding of section 102(b). It then had nothing left to compare.

DRAFT November 17, 1995 2:43 pm

interface," Pet. App. 131a, was a hierarchically organized menu structure, rather than a particular hierarchically organized menu structure. Id. 1299a-131a. Its reason for that conclusion, however, is not entirely clear.

Borland had argued that the precise hierarchy is necessary to the function of the program. The district court responded, in an analysis on which Lotus relies entirely, Br. at 39-42, that "[a]s applied to any case involving a useful article, an argument of this kind would always define the idea to incorporate all the specifics of the particular expression of that idea in the allegedly copyrightable work. Nothing would be copyrightable under this methodology of analysis." Pet. App. 130a. But, even if the specified hierarchy is unprotected, the computer program expressing that hierarchy remains protected by copyright. And if the hierarchy is idea, it is unprotected under Section 102(b) regardless of how little that leaves protected.^{23/}

In our view, this abstract set of rules governing the creation and interpretation of instructions by both humans and computers, a set of rules capable of expression in many different forms, should be considered idea rather than expression. Rather

^{23/}The district court may have held the menu command hierarchy to be protected expression because Lotus could have chosen a different menu command hierarchy. Pet. App. 131a-133a. This rationale is also unpersuasive. At every level of abstraction there can be alternatives and choices; their existence is plainly not sufficient to create protectable expression. There were alternatives to a hierarchical command structure also, as there are to any idea.

DRAFT November 17, 1995 2:43 pm

than "the expression adopted by the programmer [which] is the copyrightable element in a computer program," the set of rules, like an algorithm, is one of "the actual processes or methods combined in the program [which] are not within the scope of the copyright law." H.R. Rep. No. 1476, at 57(?), reprinted in [USCode etc.] at 5670. It is the "art" that Baker v. Selden made clear is unprotected despite copyright protection for a work embodying that art.

Viewing this set of rules as idea conforms to what has been a widely held understanding that languages as such are not copyrightable. See, e.g., Stern, *supra*, at 322 ("Until quite recently few observers would have considered copyright protection for computer programming language to be a matter of legal controversy, or even concern. The general assumption was that computer programming languages are not subject to copyright protection because they were unprotectable 'ideas,' rather than protectable 'expressions' of ideas.") (footnotes omitted); Elizabeth G. Lowry, Note, Copyright Protection for Computer Languages: Creative Incentive or Technological Threat?, 39 Emory L.J. 1294, 1294 (1990) ("until 1987 no one had ever seriously considered claiming ownership to a computer language").

Industry practice has long conformed to this understanding. Books setting forth computer languages are common. For example, B. Kernighan & D. Ritchie, *The C Programming Language* (1978),

DRAFT November 17, 1995 2:43 pm

sets forth the C programming language. As Stern observes (supra, at 352):

The art of programming in C is a nonliteral element of the Kernighan and Ritchie book only to the extent that the art of Seldenian bookkeeping is a nonliteral element of Selden's book. The art of programming in the C language can be a protected nonliteral element of their book only by overruling the doctrine of Baker v. Selden.^{24/}

To program in C, one must know the defining elements of the C language, that is, its vocabulary, syntax, and semantics; the book details those. Use of the C language to instruct a computer requires a means of translating statements in the C language -- source code -- into statements the computer can understand. Creating the means to accomplish that translation by incorporating the defining elements of the language into a translation device (normally a computer program called a "compiler") cannot, under Baker, infringe the copyright on the book, for that is necessary to practice the art. Thus, anyone would be free to write a C compiler despite the copyright in a book setting forth the C language; and many have done so.^{25/} And

^{24/}We recognize that there likely were earlier embodiments of the C language in writings, just as, we suspect, there were written embodiments of the Lotus command hierarchy prior to emergence of the full-blown Lotus 1-2-3. This historical detail should not affect the force of the illustration.

^{25/}Thus, for example, a catalog offered for sale to computer programmers C compilers and interpreters for the DOS operating system from nine publishers and Basic, C++, Fortran, and Pascal compilers and interpreters from six each. Catalog, Programmer's Paradise, Fall/Winter 1993, at 36, 38, 43-44, 46. [NOTE: Finding this catalog to verify the cites may be impossible. Recent catalogs from the same company are less helpful. The following

DRAFT November 17, 1995 2:43 pm

a copyright on the first C compiler, which necessarily contains within it an expression of the rules of C, similarly is understood to confer no protection for those rules, the language itself.^{26/} Copyright on either literary work, the book or the computer program, leaves the language unprotected.

Lower court copyright decisions do not directly address the copyrightability of languages as such, but they provide some support for viewing the rules of a language as uncopyrightable idea.^{27/} For example, it has long been established that systems

is based on another catalog I do have available: Thus, for example, one catalog offers for sale to computer programmers compilers and interpreters for the C (or very similar C++) language from eight publishers, Basic compilers or interpreters from four, and Fortran compilers or interpreters from five. Catalog, The Programmer's Shop, Spring 1994, at 86-87,89. Many available compilers and interpreters are not listed.

^{26/}Cf. U.S. Copyright Office, Compendium II: Compendium of Copyright Office Practices 300-26, 300-32 (1984) (office will refuse registration where claim is based only on a programming language).

^{27/}Judge Keeton noted in Paperback that the defendant had cited no precedent supporting the contention that languages are not copyrightable. Pet. App. 244a. But there are also no prior cases holding that languages are copyrightable. Judge Keeton referred to Reiss v. National Quotation Bureau, Inc., 276 F. 717 (S.D.N.Y. 1921), but Reiss considered only the very different question whether a list of meaningless coined words, with no syntax, was a "writing."

Petitioner contends that it has long been established that "commercial cable and telegraph codes . . . were copyrightable, when embodied in the tangible medium of code books necessary to decipher their meaning," Br. 24-25, but the cases it cites, id. at 25 n.37, do not support the copyrightability of languages. Hartfield v. Peterson, 91 F.2d 998 (2d Cir. 1937), apparently involved a book listing code phrases, arranged alphabetically under certain headings. Id. at 999. No issue of the copyrightability of such books, if original, was raised. The

DRAFT November 17, 1995 2:43 pm

of shorthand are not copyrightable, although works explaining the use of such systems may be protected. Brief English Systems v. Owen, 48 F.2d 555 (2d Cir.), cert. denied, 283 U.S. 858 (1931); Griggs v. Perrin, 49 F. 15 (N.D.N.Y. 1892).^{21/} As the Second Circuit explained in Brief English Systems, "[t]here is no literary merit in a mere system of condensing written words into less than the number of letters usually used to spell them out. Copyrightable material is found, if at all, in the explanation of how to do it."^{22/}

issue was whether the defendant had copied from plaintiff's book, or whether the similarities resulted from the use of common sources. The court, treating plaintiff's book as a compilation, emphasized that the compilation copyright protected the whole work, id. at 1000, and defendant was not free to copy from it. There is no indication that the court intended to protect rules of encoding and decoding, or the structure of a language. In American Code Co. v. Benginger, 232 F. 829 (2d Cir. 1922), plaintiff claimed to have added a column of code words to a work uncopyrighted in the United States, and the court found the list copyrightable. Id. at 833. It saw little difficulty in preliminarily enjoining defendant's distribution of photolithographic copies of plaintiff's book. Hartfield v. Herzfeld, 60 F. 599 (S.D.N.Y. 1932), is even less on point. In that case, the defendant waived the question of infringement, and the only issue before the court was whether the plaintiff had authorized defendant's copying. None of these cases refers to the idea/expression dichotomy.

^{21/}Shorthand systems typically borrow the grammar and syntax of the language they are used to represent, and so the shorthand cases may not be exactly on point. However, at a minimum they suggest that the notational conventions selected for a language, such as the letters marking positions in the Lotus language, are not copyrightable.

^{22/}In Synercom Tech. v. University Computing Co., 462 F. Supp. 1003, 1012-1014 (N.D. Tex. 1978), defendant's computer program accepted and used data stored in the format of plaintiff's copyrighted format cards. The court held that the ordering and sequencing of the data was idea, not expression.

DRAFT November 17, 1995 2:43 pm

Courts also consider other sets of rules, such as the rules of games, to be unprotectable idea, although particular expressions of those rules, and the actual implementation of those rules in playable games, may be protectable. Thus in Atari, Inc. v. North American Philips Consumer Electronics Corp., 672 F.2d 607 (7th Cir.), cert. denied, 459 U.S. 880 (1982), the court considered whether K.C. Munchkin infringed the (audio-visual) copyright on PAC-MAN. In light of the idea-expression dichotomy of Section 102(b), the court concluded that "copyright protection does not extend to games as such." Id. at 615. It found that PAC-MAN "can be described accurately in fairly abstract terms, much in the same way as one would articulate the rules to such a game," holding that "[t]he audio component and the concrete details of the visual presentation constitute the

Since ordering and sequencing are the rules that give meaning to digits punched in cards, Synercom implicitly holds a language to be uncopyrightable idea. Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994), pet. for reh'g en banc denied, 46 F.3d 408 (5th Cir. 1995), rejected aspects of Synercom, but not this one. Engineering Dynamics involved the very formats at issue in Synercom, along with others, 26 F.3d at 1339, but no copyright protection for individual formats was claimed, only protection for the sequence and organization of the formats as a whole. Id. The court, emphasizing that the formats are "quasi-textual [and] consist of a series of words and a framework of instructions that act as prompts for the insertion of relevant data," id. at 1342, 1344, reversed a finding that the formats were unprotectable and remanded for further determinations. This suggests that the court would not protect a bare language. The court later explained, 46 F.23 at 410, that its opinion "cannot properly be read to extend . . . to the practice employed by users of programs of analyzing application programs to 'read' the file formats of other programs." This explanation is consistent with the Synercom holding discussed above.

DRAFT November 17, 1995 2:43 pm

copyrightable expression of that game 'idea.'" Id. at 617. Accord, M. Kramer Mfg. Co. v. Andrews, 783 F.2d 521, 435 (4th Cir. 1986) ("[s]trictly speaking, the game, the idea of the game, itself is not protected"). See also Morrissey v. Procter & Gamble, Inc., 379 F.2d 675, 678 (1st Cir. 1967) (substance of sweepstakes contest not copyrightable). Cf. Crume v. Pacific Mut. Life Ins. Co., 140 F.2d 182 (7th Cir. 1944) (holder of copyright on pamphlets disclosing form of reorganization plan recognizes defendant's right to use the plan, claiming infringement only as to words used).

The rules that allow communication with a computer in the Lotus language, like the rules that allow the playing of a particular game or the practice of a particular accounting system, are abstract ideas that copyright does not remove from the public domain. Though they may be expressed in copyrightable form, they are not themselves copyrightable expression under section 102(b).

3. Preserving the public's right freely to use the rules defining the structure of a computer language to create original expression serves the fundamental policy considerations underlying the Copyright Act. "[T]he Copyright Act must be construed in light of [its] basic purpose" of "stimulat[ing] artistic creativity for the general public good." Twentieth Century Music Corp., 422 U.S. at 156. Ideas are unprotected by copyright because "[t]o grant property status to a mere idea

DRAFT November 17, 1995 2:43 pm

would permit withdrawing the idea from the stock of materials that would otherwise be open to other authors, thereby narrowing the field of thought open for development and exploitation." Nimmer, supra, § 13.03[B], at 13-70.

In distinguishing idea from expression, "the line must be a pragmatic one, which also keeps in consideration 'the preservation of the balance between competition and protection reflected in the patent and copyright laws.'" Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983) (citation omitted), cert. dismissed, 464 U.S. 1033 (1984), quoted in part in Altai, 982 F.2d at 711. Thus, the Second Circuit has observed that "the importance of keeping ideas from private ownership is far greater for ideas . . . directed to the understanding of phenomena or the solving of problems, than for those that merely represent the author's taste or opinion and therefore do not materially assist the understanding of future thinkers," CCC Information Serv. v. Maclean Hunter Mkt. Rep., 44 F.3d 61, 71 (2d Cir. 1994). In its view, section 102(b) "contemplates denying protection to building-block ideas explaining processes or discoveries" rather than "subjective opinion," id. at 71 n.22,

Languages, in the world of computers as well as elsewhere, are just such building blocks. They are essential elements in the use of computers. The rules of language dictate the manner in which humans communicate commands to computer programs,

DRAFT November 17, 1995 2:43 pm

whether interactively at the keyboard or through macros from the short and simple to the long and complex. They dictate the form in which data processed by computer programs are stored and provided to the programs. See note 29 supra. Compilers and interpreters transform computer programs in the form of source code written in "higher level" languages (such as C, Pascal, Fortran, and Basic) into object code in machine language, see, e.g., Apple Computer, 714 F.2d at 1243; the higher level language is in effect a command language that tells the compiler what object code to produce. The compiler or interpreter must incorporate the rules of the language in order to function. Operating systems "manage the internal functions of the computer or facilitate use of application programs." Franklin, 714 F.2d at 1240. Applications programs must communicate with operating systems. Thus, "[o]perating systems establish standard protocols and formats to which application programs . . . must be tailored." Peter S. Menell, Tailoring Legal Protection for Computer Software, 39 Stan. L. Rev. 1329, 1345 (1987). These protocols amount to a language applications programs use to cause operating systems to perform functions.²⁹ Again, the rules of language must be incorporated.

²⁹Cf. Altai, 982 F.2d at 698-99 (function of subprogram at issue "is to translate the language of a given program into the particular language that the computer's own operating system can understand"; requests to operating system for resources made through use of "system calls").

DRAFT November 17, 1995 2:43 pm

The competitive consequences of construing the copyright law to protect the structure of a language, may be substantial, because languages are important building blocks. As the district court recognized, "[d]rawing the line too liberally in favor of copyright protection would bestow strong monopolies over specific applications upon the first to write programs performing those applications and would thereby inhibit other creators from developing improved products." Pet. App. 206a (quoting Peter S. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 Stan. L. Rev. 1045, 1047-48 (1989)); cf. Altai, 982 F.2d at 712. See also Pet. App. 20a-21a (majority), 26a-27a (Boudin, J., concurring). Copyright protection for language structure tends to Balkanize the world of computers by preventing communication based on common building blocks.

In enacting section 102(b) of the Copyright Act, Congress made the choice to place limits on copyright protection in order to promote the free exchange and wide availability of ideas. Interpreting section 102(b) to deny Lotus a copyright on the rules themselves, while protecting its rights to preclude copying of its program, is faithful to the language Congress chose and its purposes.

CONCLUSION

This Court should affirm the judgment below.

Order -

filing out-of-time
in support of Co two -

SG would have
major objection

if do it on same day -
motion will be denied -
because Boreland has
no chance to respond.

have to do it
someplace in
between

E X E C U T I V E O F F I C E O F T H E P R E S I D E N T

01-Dec-1995 01:38pm

TO: Jack M. Quinn
TO: Elena Kagan

FROM: Kathleen M. Wallman
 Office of the Counsel

SUBJECT: copyright matter

JQ & EK -- I spoke with Bo; he will arrange the NEC meeting that Ginger Lew requested on Monday. He understands that DOJ would like guidance on Monday, but asked us to obtain a day's leeway from them because NEC's staff will be preoccupied with budget matters and cannot arrange the meeting before Monday.

I propose that Elena communicate with Joel and David O. and tell them that we will provide guidance as soon as possible, and by Tuesday at the latest. Jack, do you want to call Jamie, or do you want me to call Seth or John so that we are responding also to people who were at the meeting? KW

LESTER S. HYMAN
OF COUNSEL

SWIDLER
&
BERLIN
CHARTERED

DIRECT DIAL
(202)424-7509

November 27, 1995

VIA FACSIMILE & HAND DELIVERY

The Honorable John M. Quinn
Counsel to the President
The White House
2nd Floor, West Wing
1600 Pennsylvania Avenue, N.W.
Washington, D.C. 20500

Re: Lotus Development Corp. v. Borland International, Inc.

Dear Jack:

We write on behalf of our client, Lotus Development Corporation, to request that you urge the Solicitor General not to file an amicus brief undermining copyright protection for computer software in the above Supreme Court case. This case is important; the U.S. computer software industry is one of our strongest forces in the global economy and the copyright protection that helps make it so must be preserved. A decision weakening this protection would undermine Administration efforts to strengthen the copyright protections U.S. exporters receive abroad.

The Supreme Court granted certiorari to review a decision of the First Circuit Court of Appeals holding that the "menu command hierarchy" of a computer program is uncopyrightable, as a matter of law, no matter how creative and original it may be and despite the availability of numerous possible alternative menus. This case involves "Lotus 1-2-3," a popular spreadsheet program that allows users to perform accounting functions. The program's "menu command hierarchy" is a series of words appearing on the screen that instruct a user how to operate the program and is part of what commonly is referred to as a program's "user interface." Borland copied the entire Lotus 1-2-3 menu command hierarchy into its competing spreadsheet program in order to take advantage of the popularity 1-2-3 then enjoyed. Lotus brought an infringement suit in a Massachusetts federal district court, which held after two separate trials that Borland was liable for infringement. The First Circuit reversed, holding broadly (and wrongly) that menu command hierarchies are uncopyrightable per se. Lotus Development Corp. v. Borland Intern. Inc., 49 F.3d 807 (1st Cir.1995).

The First Circuit is the first and only court so to rule. Consistent with Congress's mandate, the courts of appeals in virtually every other circuit had reached consensus that

3000 K STREET, N.W. ■ SUITE 300
WASHINGTON, D.C. 20007-5116
(202)424-7500 ■ TELEX 701131 ■ FACSIMILE (202)424-7643

cc:
KATHY/JAMES
11-28/95

The Honorable Jack Quinn
November 27, 1995
Page 2

copyright protects the original, expressive elements of computer programs -- in both internal codes and user interfaces -- as literary works, subject to the same rules applicable to such works. The First Circuit rejected all this in an opinion that threatens meaningful copyright protection for all computer software, as the concurring judge in the Lotus case conceded, 49 F.3d at 820.

As discussed in the attachment to this letter, the First Circuit opinion is misguided. In addition, there are powerful policy reasons why the Government should not support the First Circuit's decision before the Supreme Court.

The First Circuit's ruling, if upheld, would have a devastating effect on the vital computer software programming industry. A crucial feature of a computer program is its "user interface" -- that portion of the program with which the human user interacts. The menu command hierarchy is a key feature of the user interface. Designing "user friendly" interfaces that make programs easier to learn and use (without spending hours reading a manual) is a difficult creative and competitive challenge and can involve immense investment of resources. But according to the First Circuit, user interfaces are ineligible for copyright protection.

The computer-software industry is among this country's most important exporters. In 1992, U.S. suppliers captured an estimated 74 percent of the world packaged software market.^{1/} A major policy goal of both this and the prior Administration -- a goal that has strong Congressional support^{2/} -- is to encourage other countries to strengthen copyright protection for computer programs sold in international trade. It is critical that domestic policy be congruent with the positions we urge our trading partners to adopt.

On February 28, 1990, Ambassador Hills requested that the European Community Software Directive be written in conformity with U.S. law, stating specifically that "[c]opyright in a computer program should extend to all copyrightable elements, including interfaces." (emphasis added).^{3/} On November 2, 1993, Secretary Brown and Ambassador

^{1/} U.S. Department of Commerce, U.S. Industrial Outlook 1994 at 27-5.

^{2/} For example, the Omnibus Trade and Competitiveness Act of 1988 directed the U.S. Trade Representative to identify foreign countries that "deny adequate and effective protection of intellectual property rights" and to initiate appropriate investigations. 19 U.S.C. §§2242(a)(2), 2412(b)(2).

^{3/} Letter dated February 28, 1990 to Martin Bangemann, Vice President of the Commission of the European Community.

Kantor, in a letter to Japan's Minister of International Trade and Industry, expressed "grave concerns" about an initiative to "weaken Japan's protection of computer programs under its copyright law."^{4/} Provisions strengthening protections for U.S. software also were key elements of both NAFTA and the latest GATT Agreement.^{5/} In February of this year, U.S. trade negotiators, after much effort, successfully resolved a major dispute with China over intellectual property rights, reaching an accord that Ambassador Kantor predicted would have "enormous impact" on American business.^{6/} The credibility of the Administration's continuing efforts to strengthen foreign copyright protections for U.S. computer software would be undermined if the Department of Justice now were to support a decision that significantly weakens the protection available under U.S. law.

Enforcing Lotus' full intellectual property rights in its Lotus 1-2-3 program poses no threat to competition in the computer programming industry. Quite the contrary, under the protection of a strong copyright law, there has been an unprecedented flow into the market of varied and innovative new software products, which offer dramatically improved user interfaces and greatly expanded power and functionality at lower prices. Small companies have developed many of these innovative products. Without copyright protection for user interfaces, small developers quickly could lose their markets to "software factories" that build "reverse-engineered" imitations to compete with the originals. Small companies no longer could aspire, as in the past, to become market leaders on the strength of innovative new products.

No basis exists for the concern expressed by the concurring First Circuit judge that a software developer could obtain a "lock" on the market by copyrighting the user interface of a best-selling product. Indeed, the history of this case shows the contrary. Despite Lotus' vigorous defense of its copyrights, the 1-2-3 program has lost its previous best-selling status and presently has only around ten percent of the market for spreadsheet programs. In a competitive market, customers are not locked into a best-selling program; if a competitor offers a better program with a more user-friendly interface, the current best-seller likely will

^{4/} Letter dated November 2, 1993 to the Hon. Hiroshi Kumagai, Minister of International Trade and Industry.

^{5/} Testimony of Ambassador Kantor before the House Ways and Means Committee dated January 26, 1994; BNA International Trade Reporter, "Mexico Lures High-Tech Exports by Reform of Investment, Intellectual Property Law" (September 22, 1993).

^{6/} Business Times, February 28, 1995; Dallas Morning News, February 27, 1995.

The Honorable Jack Quinn
November 27, 1995
Page 4

lose market dominance. Competitors require creativity and innovation to compete -- not the right to copy.⁷¹

The Solicitor General presently is considering whether to file a brief in the Supreme Court in support of Borland. Any such brief must be filed by December 8. We entreat you to communicate to the Solicitor General the adverse consequences a filing could have on Administration trade policies and to urge him to refrain from so acting.

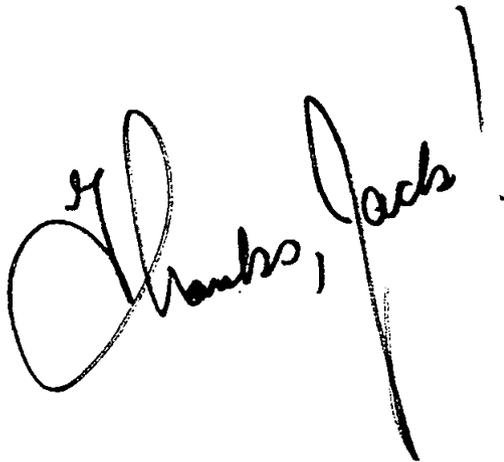
Sincerely,



Bester S. Hyman

Enclosure

LSH:dcw



Thanks, Jack!

6048759.1*

⁷¹ The attachment discusses additional competition concerns that have been raised concerning an aspect of this case.

Lotus v. Borland -- Legal Issues

1. Coverage of Copyright Act

Congress made it clear in both the Copyright Act of 1976 and the 1980 Software Amendments that copyright protection covers computer programs. The Act generally confers copyright protection on “original works of authorship,” including “literary works.” 17 U.S.C. § 102(c). Congress has defined “literary works” to include works “expressed in words, numbers, or other verbal or numerical symbols or indices, regardless of the nature of the material objects, such as books, periodicals . . . tapes, disks, or cards, in which they are embodied.” 17 U.S.C. § 101 (emphasis added). The legislative history explains that Congress intended this definition to cover “computer data bases, and computer programs.” H.R. Rep. No. 1476, 94th Cong. 2d Sess. at 54, reprinted at 1976 U.S. Code Cong. & Adm. News 5659, 5667.^{1/}

Congress’ decision to recognize copyright protection for computer programs was well within traditional copyright law, which protects useful as well as artistic works. The Constitution itself declares that the goal of copyright is to “promote the Progress of Science and useful Arts.” Art. I, § 8, cl. 8. The First Congress extended copyright protection to maps and charts,^{2/} and the courts, over the years, have recognized consistently the copyrightability of dictionaries, technical manuals, vocabulary lists, and countless other forms of utilitarian writings.^{3/}

^{1/} In 1980, Congress defined a “computer program” as “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.

^{2/} Act of May 31, 1970, ch. 15, § 1, 1 Stat. 124 (repealed 1831).

^{3/} E.g., United Dictionary Co. v. G. & C. Merriam Co., 208 U.S. 260 (1908); No-Leak-O Piston Ring Co. v. Norris, 277 F. 951 (4th Cir. 1921); College Entrance Book Co. v. Amsco Book Co., 119 F.2d 874 (2d Cir. 1941).

At this case's core is the fundamental, doctrine that copyright protects the expression of an idea, but not the idea itself. This distinction, which can be traced to the Supreme Court's decision in Baker v. Selden, 101 U.S. 99 (1879), was codified in Section 102(b) of the 1976 Act. In holding the Lotus 1-2-3 menu command hierarchy copyrightable, the district court applied the "idea/expression" dichotomy. The court, after trial, found that Borland had gone beyond copying the ideas embodied in the Lotus 1-2-3 menu command hierarchy and had copied Lotus's particular method of expressing those ideas. Lotus Development Corp. v. Borland Intern., Inc., 799 F. Supp. 203, 212-14, 831 F. Supp. 223, 231-33 (D. Mass. 1992, 1993).

The First Circuit did not question this factual finding. Instead, the First Circuit held that the Lotus 1-2-3 menu command hierarchy is a "method of operation" and therefore even its expressive element is not copyrightable. Lotus Development Corp. v. Borland Intern., Inc., 49 F.3d 807 (1st Cir. 1995). The First Circuit relied on section 102(b) of the Copyright Act, which denies copyright 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," (17 U.S.C. § 102(b)). The First Circuit, however, mistakenly concluded that a court may apply the traditional "idea/expression" dichotomy only after it finds that a work is not excluded under section 102(b). 49 F.2d at 815-16. In so doing, the court erroneously put the cart before the horse.

The First Circuit's ruling is flatly contrary to the Copyright Act. As described, both the 1976 Act and the 1980 Software Amendments reflect Congress' clear intent to extend copyright protection to computer programs.

The First Circuit's conclusion that Section 102(b) precludes copyright protection for even the separate expressive elements of menu command hierarchies eviscerates the Act's protection of computer programs, for all parts of computer programs are at bottom "methods of operation." Moreover, the legislative history of section 102(b) contradicts the First Circuit opinion, for the House Judiciary Committee explained that "[s]ection 102(b) in no way enlarges or contracts the scope of copyright protection under the present law." Instead, the Committee observed, the purpose of section 102(b) "is to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged." H.R. Rep. No. 1476, 94th Cong. 2d Sess. at 57 ("House Report"), reprinted in 1976 U.S. Code Cong. & Admin. News 5670 (emphasis added). The Committee emphasized that it wanted the "idea/expression" dichotomy applied to computer programs, commenting that copyright protection reaches computer programs "to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves." House Report at 54, 1976 U.S. Code Cong. & Admin. News 5667. The First Circuit, on the other hand, concluded that the "idea/expression" dichotomy was irrelevant to its decision.

In sum, the trial court found that Borland copied the expressive elements of a portion of Lotus' computer program. The First Circuit did not dispute that finding, but held instead that "expression that is part of a 'method of operation' cannot be copyrighted." 49 F.3d at 818. That holding is flatly contrary to the 1976 Copyright Act's extension of copyright coverage and principles -- including the "idea/expression" dichotomy -- to computer programs.

2. Apparent Antitrust Concerns

The impetus to file an amicus brief apparently comes from the Antitrust Division, which is concerned about one narrow portion of the district court's ruling involving a feature of Borland products referred to as the "Key Reader." After Lotus sued, Borland introduced the Key Reader feature as a method by which shortcut steps (called "macros") written using Lotus 1-2-3 could be used when 1-2-3 files were imported into a Borland product. The district court found infringement because "the Key Reader file contains a virtually identical [although hidden] copy of the Lotus menu tree structure," 831 F. Supp. at 228.

The antitrust concern apparently is that the district court's "Key Reader" holding, if extended, might enable a software publisher with a best-selling program to use the copyright laws to prevent competitors from designing compatible programs. This concern is misplaced, for several reasons.

a. The district court's decision was written narrowly and correctly was based upon the specific facts before the court. Lotus did not argue below, and will not before the Supreme Court, that this ruling has sweeping ramifications.

b. The designer of a complementary program seeking compatibility does not have to copy protectible expression from another program into its own to achieve this goal. Rather, the designer must shape the complementary program so it issues instructions the other program will understand. To "plug into" the best-selling program, a competitor must know the shape of the "socket," but it must not necessarily make the "socket" part of its own program.

c. Even if some minimal copying were involved, this might well be a "fair use," which copyright law protects. 17 U.S.C. § 107. The Ninth Circuit found fair use where a limited

portion of a computer program was used to provide a product that worked with and complemented, but did not supersede, the original. Lewis Galoob Toys, Inc. v. Nintendo of America, Inc., 964 F.2d 965, 972 (9th Cir. 1992). The fair use doctrine thus ameliorates antitrust concerns. Other traditional copyright defenses such as copyright misuse, also might be available, depending upon the specific facts.

d. If a dominant software publisher did violate the antitrust laws, antitrust remedies would be available (which could include an order to grant copyright licenses). The copyright protection Congress afforded to computer programs should not be eviscerated because of some unfocused fear that a dominant company might abuse its copyrights and violate the antitrust laws.

e. The history of the computer program marketplace shows that consumers prefer open systems, and that most so-called "industry standards" -- including Lotus 1-2-3 -- do not last forever. So long as competition exists in a market and consumers may choose among various attractive offerings, attempts by program manufacturers to deny or restrict compatibility may well be self-defeating and risk obsolescence.

f. The issue here is not an attempt by a manufacturer of a dominant operating system (which runs a computer) to deny or restrict compatibility regarding the application software developed by others that must run on the operating system. Here the issue is simply whether a maker of a software application may copy the expressive elements of another's application. In this context, providing meaningful copyright protection, as Congress has mandated, does not present any meaningful risk to competition or the policies protected by the antitrust laws.

David Oden -
ending

AT draft

Not SG going forward
to SG

SG working
I would deal with it -
concerns. meet other

CR long
not just AT long

Telecom David Orden 12/11

Skill hopping - way to craft in a way that's acceptable to

Antitrust

PTO/copyright office

Not a lot of optimism

Mtg last night - sides dup in

Curriculus - imp. case

- reasoning of Ist Ci - problematic
reject literal approach
vt concept idea/expression

AT - MCHs are a language -
rules of game -

DEAS -

building blocks of expression.

otherwise: establish
enormous
barriers to entry.
(new entrant - the
software program)

Always
any user
interface

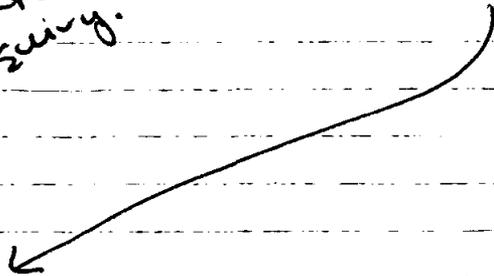
PTO - ~~Patent~~

MCHs - systems thru which ~~to~~ users communicate -

No per se rule

- Depends on originality, copying / other factors
Reverse + reward

emphasis
contextual
inspiring.



Timing problem - too late.

SG says.

Need to file w/ pabr if RTR

Could file for leave to file
out of time.

But no excuse.

Credibility of SG.

Concern w/ Ist Ci approach -

By saying we don't care

exp v. Ideas

just MPO -

call into Σ Comp's decision -

to protect programs

bec prog are method by
which comp. is operated.

Prog should be protected -

need to protect originality

1st part - everyone agrees - wrong analysis.

Lots of P. are saying This - US doesn't need to

2nd part - now trying to make narrower.

May be cases - extreme copy or extreme copying -

may be copyrightable -

poss to have CH that is protected.

Monday/noon - deadline for filing for argument.

Could file + w/draw.



Office of the Deputy Attorney General
Washington, D.C. 20530

November 30, 1995

BY HAND

Ms. Elena Kagan
Associate Counsel to the President
OEOB 125
17th St. and Pennsylvania Ave., N.W.
Washington, D.C. 20500

Dear Elena:

As you requested, enclosed are various materials relating to Lotus v. Borland. In this package, you should find Petitioner's merits brief, the cert. papers, the First Circuit's decision, and the parties' briefs in the First Circuit. I have not enclosed any of the amicus briefs from the Court of Appeals. (If you would like to have those, I would be glad to forward them, too -- they are voluminous.) The United States did not participate in the First Circuit.

As I mentioned to you, the Solicitor General's office has a policy against releasing draft briefs, so I am not in a position to provide that to you.

I hope this is helpful. If you have thoughts or questions about the case, I would be interested in hearing them. And please let me know if I can be of any further assistance.

Very truly yours,

A handwritten signature in black ink, appearing to read "D. W. Ogden", is written over the typed name.

David W. Ogden
Associate Deputy Attorney General

Enclosures (as stated)

United States Court of Appeals
For the First Circuit

No. 93-2214

LOTUS DEVELOPMENT CORPORATION,
Plaintiff, Appellee,

v.

BORLAND INTERNATIONAL, INC.,
Defendant, Appellant.

APPEAL FROM THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS
[Hon. Robert E. Keeton, U.S. District Judge]

Before

Torruella, Chief Judge,
Boudin and Stahl, Circuit Judges.

Gary L. Reback, with whom Peter N. Detkin, Michael Barclay, Isabella E. Fu, Wilson Sonsini Goodrich & Rosati, Peter E. Gelhaar, Katherine L. Parks, and Donnelly Conroy & Gelhaar, were on brief for appellant.

Matthew P. Poppel, et. al, were on brief for Computer Scientists, amicus curiae.

Dennis S. Karjala and Peter S. Menell on brief, amici curiae.

Jeffrey C. Cannon and Baker Keaton Seibel & Cannon were on brief for Computer Software Industry Association, amicus curiae.

Laureen E. McGurk, David A. Rabin, Bryan G. Harrison and Morris Manning & Martin were on brief for Chicago Computer Society, Diablo Users Group, Danbury Area Computer Society, IBM AB Users Group, Kentucky-Indiana Personal Computer Users Group, Long Island PC Users Group, Napa Valley PC Users Group, Pacific Northwest PC Users Group, Palmetto Personal Computer Club, Philadelphia Area Computer Society,

Inc., Phoenix IBM PC Users Group, Pinellas IBM PC Users Group, Quad Cities Computer Society, Quattro Pro Users Group, Sacramento PC Users Group, San Francisco PC Users Group, Santa Barbara PC Users Group, Twin Cities PC Users Group, and Warner Robbins Personal Computer Association, amici curiae.

Diane Marie O'Malley and Hanson Bridgett Marcus Vlahos & Rudy were on brief for Software Entrepreneurs' Forum, amicus curiae.

Peter M.C. Choy was on brief for American Committee for Interoperable Systems, amicus curiae.

Howard B. Abrams, Howard C. Anawalt, Stephen R. Barnett, Ralph S. Brown, Stephen L. Carter, Amy B. Cohen, Paul J. Heald, Peter A. Jaszi, John A. Kidwell, Edmund W. Kitch, Roberta R. Kwall, David L. Lange, Marshall Leaffer, Jessica D. Litman, Charles R. McManis, L. Ray Patterson, Jerome H. Reichman, David A. Rice, Pamela Samuelson, David J. Seipp, David E. Shipley, Lionel S. Sobel, Alfred C. Yen, and Diane L. Zimmerman were on brief for Copyright Law Professors, amicus curiae.

Henry B. Gutman, with whom Kerry L. Konrad, Joshua H. Epstein, Kimberly A. Caldwell, O'Sullivan Graev & Karabell, Thomas M. Lemberg, James C. Burling, and Hale and Dorr, were on brief for appellee.

Morton David Goldberg, June M. Besek, David O. Carson, Jesse M. Feder, Schwab Goldberg Price & Dannay, and Arthur R. Miller were on brief for Apple Computer, Inc., Digital Equipment Corporation, International Business Machines Corporation, and Xerox Corporation, amici curiae.

Jon A. Baumgarten, Proskauer Rose Goetz & Mendelsohn, and Robert A. Gorman were on brief for Adobe Systems, Inc., Apple Computer, Inc., Computer Associates International, Inc., Digital Equipment Corporation, and International Business Machines Corporation, amici curiae.

Herbert F. Schwartz, Vincent N. Palladino, Susan Progoff, Fish & Neave, William J. Cheeseman, and Foley Hoag & Eliot, were on brief for Computer and Business Equipment Manufacturers Association, amicus curiae.

March 9, 1995

STAHL, Circuit Judge. This appeal requires us to decide whether a computer menu command hierarchy is copyrightable subject matter. In particular, we must decide whether, as the district court held, plaintiff-appellee Lotus Development Corporation's copyright in Lotus 1-2-3, a computer spreadsheet program, was infringed by defendant-appellant Borland International, Inc., when Borland copied the Lotus 1-2-3 menu command hierarchy into its Quattro and Quattro Pro computer spreadsheet programs. See Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (D. Mass. 1992) ("Borland I"); Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass. 1992) ("Borland II"); Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 202 (D. Mass. 1993) ("Borland III"); Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 223 (D. Mass. 1993) ("Borland IV").

I.

Background

Lotus 1-2-3 is a spreadsheet program that enables users to perform accounting functions electronically on a computer. Users manipulate and control the program via a series of menu commands, such as "Copy," "Print," and "Quit." Users choose commands either by highlighting them on the screen or by typing their first letter. In all, Lotus 1-2-3 has 469 commands arranged into more than 50 menus and submenus.

Lotus 1-2-3, like many computer programs, allows users to write what are called "macros." By writing a macro, a user can designate a series of command choices with a single macro keystroke. Then, to execute that series of commands in multiple parts of the spreadsheet, rather than typing the whole series each time, the user only needs to type the single pre-programmed macro keystroke, causing the program to recall and perform the designated series of commands automatically. Thus, Lotus 1-2-3 macros shorten the time needed to set up and operate the program.

Borland released its first Quattro program to the public in 1987, after Borland's engineers had labored over its development for nearly three years. Borland's objective was to develop a spreadsheet program far superior to existing programs, including Lotus 1-2-3. In Borland's words, "[f]rom the time of its initial release . . . Quattro included enormous innovations over competing spreadsheet products."

The district court found, and Borland does not now contest, that Borland included in its Quattro and Quattro Pro version 1.0 programs "a virtually identical copy of the entire 1-2-3 menu tree." Borland III, 831 F. Supp. at 212 (emphasis in original). In so doing, Borland did not copy any of Lotus's underlying computer code; it copied only the words and structure of Lotus's menu command hierarchy. Borland included the Lotus menu command hierarchy in its programs to make them

compatible with Lotus 1-2-3 so that spreadsheet users who were already familiar with Lotus 1-2-3 would be able to switch to the Borland programs without having to learn new commands or rewrite their Lotus macros.

In its Quattro and Quattro Pro version 1.0 programs, Borland achieved compatibility with Lotus 1-2-3 by offering its users an alternate user interface, the "Lotus Emulation Interface." By activating the Emulation Interface, Borland users would see the Lotus menu commands on their screens and could interact with Quattro or Quattro Pro as if using Lotus 1-2-3, albeit with a slightly different looking screen and with many Borland options not available on Lotus 1-2-3. In effect, Borland allowed users to choose how they wanted to communicate with Borland's spreadsheet programs: ~~either by using menu commands designed by Borland, or by using the commands and command structure used in Lotus 1-2-3 augmented by Borland-added commands.~~

Lotus filed this action against Borland in the District of Massachusetts on July 2, 1990, four days after a district court held that the Lotus 1-2-3 "menu structure, taken as a whole -- including the choice of command terms [and] the structure and order of those terms," was protected expression covered by Lotus's copyrights. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 68, 70 (D. Mass. 1990)

("Paperback").¹ Three days earlier, on the morning after the Paperback decision, Borland had filed a declaratory judgment action against Lotus in the Northern District of California, seeking a declaration of non-infringement. On September 10, 1990, the district court in California dismissed Borland's declaratory judgment action in favor of this action.

Lotus and Borland filed cross motions for summary judgment; the district court denied both motions on March 20, 1992, concluding that "neither party's motion is supported by the record." Borland I, 788 F. Supp. at 80. The district court invited the parties to file renewed summary judgment motions that would "focus their arguments more precisely" in light of rulings it had made in conjunction with its denial of their summary judgment motions. Id. at 82. Both parties filed renewed motions for summary judgment on April 24, 1992. In its motion, Borland contended that the Lotus 1-2-3 menus were not copyrightable as a matter of law and that no reasonable trier of fact could find that the similarity between its products and Lotus 1-2-3 was sufficient to sustain a determination of infringement. Lotus contended in its motion that Borland had copied Lotus 1-2-3's entire user interface and had thereby infringed Lotus's copyrights.

1. Judge Keeton presided over both the Paperback litigation and this case.

On July 31, 1982, the district court denied Borland's motion and granted Lotus's motion in part. The district court ruled that the Lotus menu command hierarchy was copyrightable expression because

[a] very satisfactory spreadsheet menu tree ~~can be reconstructed using different commands and a different command structure from those of Lotus 1-2-3.~~ In fact, Borland ~~has reconstructed~~ just such an alternate tree for use in Quattro Pro's native mode. Even if one holds the arrangement of menu commands constant, it is possible to generate literally millions of satisfactory menu trees by varying the menu commands employed.

Borland II, 799 F. Supp. at 217. The district court demonstrated this by offering alternate command words for the ten commands that appear in Lotus's main menu. Id. For example, the district court stated that "[t]he 'Quit' command could be named 'Exit' without any other modifications," and that "[t]he 'Copy' command could be called 'Clone,' 'Ditto,' 'Duplicate,' 'Imitate,' 'Mimic,' 'Replicate,' and 'Reproduce,' among others." Id. Because so many variations were possible, the district court concluded that the Lotus developers' choice and arrangement of command terms, reflected in the Lotus menu command hierarchy, constituted copyrightable expression.

In granting partial summary judgment to Lotus, the district court held that Borland had infringed Lotus's copyright in Lotus 1-2-3:

[A]s a matter of law, Borland's Quattro products ~~infringe~~ the Lotus 1-2-3

copyright because of (1) the extent of copying of the "menu commands" and "menu structure" that is not genuinely disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3.

Borland II, 799 F. Supp. at 223 (emphasis in original). The court nevertheless concluded that while the Quattro and Quattro Pro programs infringed Lotus's copyright, Borland had not copied the entire Lotus 1-2-3 user interface, as Lotus had contended. Accordingly, the court concluded that a jury trial was necessary to determine the scope of Borland's infringement, including whether Borland copied the long prompts² of Lotus 1-2-3, whether the long prompts contained expressive elements, and to what extent, if any, functional constraints limited the

2. Lotus 1-2-3 utilizes a two-line menu; the top line lists the commands from which the user may choose, and the bottom line displays what Lotus calls its "long prompts." The long prompts explain, as a sort of "help text," what the highlighted menu command will do if entered. For example, the long prompt for the "Worksheet" command displays the submenu that the "Worksheet" command calls up; it reads "Global, Insert, Delete, Column, Erase, Titles, Window, Status, Page." The long prompt for the "Copy" command explains what function the "Copy" command will perform: "Copy a cell or range of cells." The long prompt for the "Quit" command reads, "End 1-2-3 session (Have you saved your work?)."

Prior to trial, the parties agreed to exclude the copying of the long prompts from the case; Lotus agreed not to contend that Borland had copied the long prompts, Borland agreed not to argue that it had not copied the long prompts, and both sides agreed not to argue that the issue of whether Borland had copied the long prompts was material to any other issue in the case. See Borland III, 831 F. Supp. at 208.

number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation. See Borland III, 831 F. Supp. at 207. Additionally, the district court granted Lotus summary judgment on Borland's affirmative defense of waiver, but not on its affirmative defenses of laches and estoppel. Borland II, 799 F. Supp. at 222-23.

Immediately following the district court's summary judgment decision, Borland removed the Lotus Emulation Interface from its products. Thereafter, Borland's spreadsheet programs no longer displayed the Lotus 1-2-3 menus to Borland users, and as a result Borland users could no longer communicate with Borland's programs as if they were using a more sophisticated version of Lotus 1-2-3. Nonetheless, Borland's programs continued to be partially compatible with Lotus 1-2-3, for Borland retained what it called the "Key Reader" in its Quattro Pro programs. Once turned on, the Key Reader allowed Borland's programs to understand and perform some Lotus 1-2-3 macros.³ With the Key Reader on, the Borland programs used Quattro Pro menus for display, interaction, and macro execution, except when they encountered a slash ("/") key in a macro (the starting key for any Lotus 1-2-3 macro), in which case they interpreted the macro as having been written

3. Because Borland's programs could no longer display the Lotus menu command hierarchy to users, the Key Reader did not allow debugging or modification of macros, nor did it permit the execution of most interactive macros.

for Lotus 1-2-3. Accordingly, people who wrote or purchased macros to shorten the time needed to perform an operation in Lotus 1-2-3 could still use those macros in Borland's programs.⁴ The district court permitted Lotus to file a supplemental complaint alleging that the Key Reader infringed its copyright.

The parties agreed to try the remaining liability issues without a jury. The district court held two trials, the Phase I trial covering all remaining issues raised in the original complaint (relating to the Emulation Interface) and the Phase II trial covering all issues raised in the supplemental complaint (relating to the Key Reader). At the Phase I trial, there were no live witnesses, although considerable testimony was presented in the form of affidavits and deposition excerpts. The district court ruled upon evidentiary objections counsel interposed. At the Phase II trial, there were two live witnesses, each of whom demonstrated the programs for the district court.

After the close of the Phase I trial, the district court permitted Borland to amend its answer to include the affirmative defense of "fair use." Because Borland had presented all of the evidence supporting its fair-use defense during the Phase I trial, but Lotus had not presented any

⁴. See Borland IV, 831 F. Supp. at 226-27, for a more detailed explanation of macros and the Key Reader.

evidence on fair use (as the defense had not been raised before the conclusion of the Phase I trial), the district court considered Lotus's motion for judgment on partial findings of fact. See Fed. R. Civ. P. 52(c). The district court held that Borland had failed to show that its use of the Lotus 1-2-3 menu command hierarchy in its Emulation Interface was a fair use. See Borland III, 831 F. Supp. at 208.

In its Phase I-trial decision, the district court found that "each of the Borland emulation interfaces contains a virtually identical copy of the 1-2-3 menu tree and that the 1-2-3 menu tree is capable of a wide variety of expression." Borland III, 831 F. Supp. at 218. The district court also rejected Borland's affirmative defenses of laches and estoppel. Id. at 218-23.

In its Phase II-trial decision, the district court found that Borland's Key Reader file included "a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names." Borland IV, 831 F. Supp. at 228. In other words, Borland's programs no longer included the Lotus command terms, but only their first letters. The district court held that "the Lotus menu structure, organization, and first letters of the command names . . . constitute part of the protectable expression found in [Lotus 1-2-3]." Id. at 233. Accordingly, the district

court held that with its Key Reader, Borland had infringed Lotus's copyright. Id. at 245. The district court also rejected Borland's affirmative defenses of waiver, laches, estoppel, and fair use. Id. at 235-45. The district court then entered a permanent injunction against Borland, id. at 245, from which Borland appeals.

This appeal concerns only Borland's copying of the Lotus menu command hierarchy into its Quattro programs and Borland's affirmative defenses to such copying. Lotus has not cross-appealed; in other words, Lotus does not contend on appeal that the district court erred in finding that Borland had not copied other elements of Lotus 1-2-3, such as its screen displays.

II.

Discussion

On appeal, Borland does not dispute that it factually copied the words and arrangement of the Lotus menu command hierarchy. Rather, Borland argues that it "lawfully copied the unprotectable menus of Lotus 1-2-3." Borland contends that the Lotus menu command hierarchy is not copyrightable because it is a system, method of operation, process, or procedure foreclosed from protection by 17 U.S.C. § 102(b). Borland also raises a number of affirmative defenses.

A. Copyright Infringement Generally

To establish copyright infringement, a plaintiff must prove "(1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original." Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 361 (1991); see also Data Gen. Corp. v. Grumman Sys. Support Corp., 36 F.3d 1147, 1160 n.19 (1st Cir. 1994); Concrete Mach. Co. v. Classic Lawn Ornaments, Inc., 843 F.2d 600, 605 (1st Cir. 1988). To show ownership of a valid copyright and therefore satisfy Feist's first prong, a plaintiff must prove that the work as a whole is original and that the plaintiff complied with applicable statutory formalities. See Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1340 (5th Cir. 1994). "In judicial proceedings, a certificate of copyright registration constitutes prima facie evidence of copyrightability and shifts the burden to the defendant to demonstrate why the copyright is not valid." Bibbero Sys., Inc. v. Colwell Sys., Inc., 893 F.2d 1104, 1106 (9th Cir. 1990); see also 17 U.S.C. § 410(c); Folio Impressions, Inc. v. Byer California, 937 F.2d 759, 763 (2d Cir. 1991) (presumption of validity may be rebutted).

To show actionable copying and therefore satisfy Feist's second prong, a plaintiff must first prove that the alleged infringer copied plaintiff's copyrighted work as a factual matter; to do this, he or she may either present direct

evidence of factual copying or, if that is unavailable, evidence that the alleged infringer had access to the copyrighted work and that the offending and copyrighted works are so similar that the court may infer that there was factual copying (i.e., probative similarity). Engineering Dynamics, 26 F.3d at 1340; see also Concrete Mach., 843 F.2d at 606. The plaintiff must then prove that the copying of copyrighted material was so extensive that it rendered the offending and copyrighted works substantially similar. See Engineering Dynamics, 26 F.3d at 1341.

In this appeal, we are faced only with whether the Lotus menu command hierarchy is copyrightable subject matter in the first instance, for Borland concedes that Lotus has a valid copyright in Lotus 1-2-3 as a whole⁵ and admits to factually copying the Lotus menu command hierarchy. As a result, this appeal is in a very different posture from most copyright-infringement cases, for copyright infringement generally turns on whether the defendant has copied protected expression as a

5. Computer programs receive copyright protection as "literary works." See 17 U.S.C. § 102(a)(1) (granting protection to "literary works") and 17 U.S.C. § 101 (defining "literary works" as "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, phonorecords, film, tapes, disks, or cards, in which they are embodied" (emphasis added)); see also H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5667 ("The term 'literary works' . . . includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.").

factual matter. Because of this different posture, most copyright-infringement cases provide only limited help to us in deciding this appeal. This is true even with respect to those copyright-infringement cases that deal with computers and computer software.

B. Matter of First Impression

Whether a computer menu command hierarchy constitutes copyrightable subject matter is a matter of first impression in this court. While some other courts appear to have touched on it briefly in dicta, see, e.g., Autoskill, Inc. v. National Educ. Support Sys., Inc., 994 F.2d 1476, 1495 n.23 (10th Cir.), cert. denied, 114 S. Ct. 307 (1993), we know of no cases that deal with the copyrightability of a menu command hierarchy standing on its own (i.e., without other elements of the user interface, such as screen displays, in issue). Thus we are navigating in uncharted waters.

Borland vigorously argues, however, that the Supreme Court charted our course more than 100 years ago when it decided Baker v. Selden, 101 U.S. 99 (1879). In Baker v. Selden, the Court held that Selden's copyright over the textbook in which he explained his new way to do accounting did not grant him a monopoly on the use of his accounting system.⁶ Borland argues:

6. Selden's system of double-entry bookkeeping is the now almost-universal T-accounts system.

The facts of Baker v. Selden, and even the arguments advanced by the parties in that case, are identical to those in this case. The only difference is that the "user interface" of Selden's system was implemented by pen and paper rather than by computer.

To demonstrate that Baker v. Selden and this appeal both involve accounting systems, Borland even supplied this court with a video that, with special effects, shows Selden's paper forms "melting" into a computer screen and transforming into Lotus 1-2-3.

We do not think that Baker v. Selden is nearly as analogous to this appeal as Borland claims. Of course, Lotus 1-2-3 is a computer spreadsheet, and as such its grid of horizontal rows and vertical columns certainly resembles an accounting ledger or any other paper spreadsheet. Those grids, however, are not at issue in this appeal for, unlike Selden, Lotus does not claim to have a monopoly over its accounting system. Rather, this appeal involves Lotus's monopoly over the commands it uses to operate the computer. Accordingly, this appeal is not, as Borland contends, "identical" to Baker v. Selden.

C. Altai

Before we analyze whether the Lotus menu command hierarchy is a system, method of operation, process, or procedure, we first consider the applicability of the test the Second Circuit set forth in Computer Assoc. Int'l, Inc. v.

Altai, Inc., 982 F.2d 693 (2d Cir. 1992).⁷ The Second Circuit designed its Altai test to deal with the fact that computer programs, copyrighted as "literary works," can be infringed by what is known as "nonliteral" copying, which is copying that is paraphrased or loosely paraphrased rather than word for word. See id. at 701 (citing nonliteral-copying cases); see also 3 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 13.03[A][1] (1993). When faced with nonliteral-copying cases, courts must determine whether similarities are due merely to the fact that the two works share the same underlying idea or whether they instead indicate that the second author copied the first author's expression. The Second Circuit designed its Altai test to deal with this situation in the computer context, specifically with whether one computer program copied nonliteral expression from another program's code.

The Altai test involves three steps: abstraction, filtration, and comparison. The abstraction step requires courts to "dissect the allegedly copied program's structure and isolate each level of abstraction contained within it." Altai, 982 F.2d at 707. This step enables courts to identify the appropriate framework within which to separate protectable

7. We consider the Altai test because both parties and many of the amici focus on it so heavily. Borland, in particular, is highly critical of the district court for not employing the Altai test. Borland does not, however, indicate how using that test would have been dispositive in Borland's favor. Interestingly, Borland appears to contradict its own reasoning at times by criticizing the applicability of the Altai test.

expression from unprotected ideas. Second, courts apply a "filtration" step in which they examine "the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain." Id. Finally, courts compare the protected elements of the infringed work (i.e., those that survived the filtration screening) to the corresponding elements of the allegedly infringing work to determine whether there was sufficient copying of protected material to constitute ~~infringement~~. Id. at 710.

In the instant appeal, we are not confronted with alleged nonliteral copying of computer code. Rather, we are faced with Borland's deliberate, literal copying of the Lotus menu command hierarchy. Thus, we must determine not whether nonliteral copying occurred in some amorphous sense, but rather whether the literal copying of the Lotus menu command hierarchy constitutes copyright infringement.

While the Altai test may provide a useful framework for assessing the alleged nonliteral copying of computer code, we find it to be of little help in assessing whether the literal copying of a menu command hierarchy constitutes copyright infringement. In fact, we think that the Altai test

in this context may actually be misleading because, in instructing courts to abstract the various levels, it seems to encourage them to find a base level that includes copyrightable subject matter that, if literally copied, would make the copier liable for copyright infringement.⁸ While that base (or literal) level would not be at issue in a nonliteral-copying case like Altai, it is precisely what is at issue in this appeal. We think that abstracting menu command hierarchies down to their individual word and menu levels and then filtering idea from expression at that stage, as both the Altai and the district court tests require, obscures the more fundamental question of whether a menu command hierarchy can be copyrighted at all. ~~The initial inquiry should not be whether individual components of a menu command hierarchy are expressive, but rather whether the menu command hierarchy as a whole can be copyrighted.~~ BUT see Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823 (10th Cir. 1993) (endorsing Altai's abstraction-filtration-comparison test as a way of determining whether "menus and sorting criteria" are copyrightable).

8. We recognize that Altai never states that every work contains a copyrightable "nugget" of protectable expression. Nonetheless, the implication is that for literal copying, "it is not necessary to determine the level of abstraction at which similarity ceases to consist of an 'expression of ideas,' because literal similarity by definition is always a similarity as to the expression of ideas." 3 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 13.03[A](2) (1993).

D. The Lotus Menu Command Hierarchy: A "Method of Operation"

Borland argues that the Lotus menu command hierarchy is uncopyrightable because it is a system, method of operation, process, or procedure foreclosed from copyright protection by 17 U.S.C. § 102(b). Section 102(b) states: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." Because we conclude that the Lotus menu command hierarchy is a method of operation, we do not consider whether it could also be a system, process, or procedure.

We think that "method of operation," as that term is used in § 102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer. Thus a text describing how to operate something would not extend copyright protection to the method of operation itself; other people would be free to employ that method and to describe it in their own words. Similarly, if a new method of operation is used rather than described, other people would still be free to employ or describe that method.

We hold that the Lotus menu command hierarchy is an uncopyrightable "method of operation." The Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3. If users wish to copy material, for example, they

use the "Copy" command. If users wish to print material, they use the "Print" command. Users must use the command terms to tell the computer what to do. Without the menu command hierarchy, users would not be able to access and control, or indeed make use of, Lotus 1-2-3's functional capabilities.

The Lotus menu command hierarchy does not merely explain and present Lotus 1-2-3's functional capabilities to the user; it also serves as the method by which the program is operated and controlled. The Lotus menu command hierarchy is different from the Lotus long prompts, for the long prompts are not necessary to the operation of the program; users could operate Lotus 1-2-3 even if there were no long prompts.⁹ The Lotus menu command hierarchy is also different from the Lotus screen displays, for users need not "use" any expressive aspects of the screen displays in order to operate Lotus 1-2-3; because the way the screens look has little bearing on how users control the program, the screen displays are not part of Lotus 1-2-3's "method of operation."¹⁰ The Lotus menu command

9. As the Lotus long prompts are not before us on appeal, we take no position on their copyrightability, although we do note that a strong argument could be made that the brief explanations they provide "merge" with the underlying idea of explaining such functions. See Morrissey v. Procter & Gamble Co., 379 F.2d 675, 678-79 (1st Cir. 1967) (when the possible ways to express an idea are limited, the expression "merges" with the idea and is therefore uncopyrightable; when merger occurs, identical copying is permitted).

10. As they are not before us on appeal, we take no position on whether the Lotus 1-2-3 screen displays constitute original expression capable of being copyrighted.

hierarchy is also different from the underlying computer code, because while code is necessary for the program to work, its precise formulation is not. In other words, to offer the same capabilities as Lotus 1-2-3, Borland did not have to copy Lotus's underlying code (and indeed it did not); to allow users to operate its programs in substantially the same way, however, Borland had to copy the Lotus menu command hierarchy. Thus the Lotus 1-2-3 code is not a uncopyrightable "method of operation."¹¹

The district court held that the Lotus menu command hierarchy, with its specific choice and arrangement of command terms, constituted an "expression" of the "idea" of operating a computer program with commands arranged hierarchically into menus and submenus. Borland II, 799 F. Supp. at 216. Under the district court's reasoning, Lotus's decision to employ hierarchically arranged command terms to operate its program could not foreclose its competitors from also employing hierarchically arranged command terms to operate their programs, but it did foreclose them from employing the specific command terms and arrangement that Lotus had used. In effect,

Is this what we're saying? must be clear.

11. Because the Lotus 1-2-3 code is not before us on appeal, we take no position on whether it is copyrightable. We note, however, that original computer codes generally are protected by copyright. See, e.g., Altai, 982 F.2d at 702 ("It is now well settled that the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection.") (citing cases).

the district court limited Lotus 1-2-3's "method of operation" to an abstraction.

Accepting the district court's finding that the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms, we nonetheless hold that that expression is not copyrightable because it is part of Lotus 1-2-3's "method of operation." We do not think that "methods of operation" are limited to abstractions; rather, they are the means by which a user operates something. If specific words are essential to operating something, then they are part of a "method of operation" and, as such, are unprotectable. This is so whether they must be highlighted, typed in, or even spoken, as computer programs no doubt will soon be controlled by spoken words.

The fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the question of whether it is a "method of operation." In other words, our initial inquiry is not whether the Lotus menu command hierarchy incorporates any expression.¹² Rather, our initial inquiry is whether the Lotus menu command hierarchy is a "method of operation." Concluding, as we do, that users operate Lotus 1-2-3 by using the Lotus menu command hierarchy, and that the entire Lotus menu command hierarchy is essential

12. We think that the Altai test would contemplate this being the initial inquiry.

too operating Lotus 1-2-3, we do not inquire further whether that method of operation could have been designed differently. These "expressive" choices of what to name the command terms and how to arrange them do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter.

Our holding that "methods of operation" are not limited to mere abstractions is bolstered by Baker v. Selden. In Baker, the Supreme Court explained that

the teachings of science and the rules and methods of useful art have their final end in application and use; and this application and use are what the public derive from the publication of a book which teaches them. . . . The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself. The object of the one is explanation; the object of the other 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.

Baker v. Selden, 101 U.S. at 104-05. Lotus wrote its menu command hierarchy so that people could learn it and use it. Accordingly, it falls squarely within the prohibition on copyright protection established in Baker v. Selden and codified by Congress in § 102(b).

In many ways, the Lotus menu command hierarchy is like the buttons used to control, say, a video cassette recorder ("VCR"). A VCR is a machine that enables one to watch and record video tapes. Users operate VCRs by pressing a

series of buttons that are typically labelled "Record, Play, Reverse, Fast Forward, Pause, Stop/Eject." That the buttons are arranged and labeled does not make them a "literary work," nor does it make them an "expression" of the abstract "method of operating" a VCR via a set of labeled buttons. Instead, the buttons are themselves the "method of operating" the VCR.

When a Lotus 1-2-3 user chooses a command, either by highlighting it on the screen or by typing its first letter, he or she effectively pushes a button. Highlighting the "Print" command on the screen, or typing the letter "P," is analogous to pressing a VCR button labeled "Play."

Just as one could not operate a buttonless VCR, it would be impossible to operate Lotus 1-2-3 without employing its menu command hierarchy. Thus the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves. Unlike the labels on a VCR's buttons, which merely make operating a VCR easier by indicating the buttons' functions, the Lotus menu commands are essential to operating Lotus 1-2-3. Without the menu commands, there would be no way to "push" the Lotus buttons, as one could push unlabeled VCR buttons. While Lotus could probably have designed a user interface for which the command terms were mere labels, it did not do so here. Lotus 1-2-3 depends for its operation on use of the precise command terms that make up the Lotus menu command hierarchy.

One might argue that the buttons for operating a VCR are not analogous to the commands for operating a computer program because VCRs are not copyrightable, whereas computer programs are. VCRs may not be copyrighted because they do not fit within any of the § 102(a) categories of copyrightable works; the closest they come is "sculptural work." Sculptural works, however, are subject to a "useful-article" exception whereby "the design of a useful article . . . shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article." 17 U.S.C. § 101. A "useful article" is "an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." Id. Whatever expression there may be in the arrangement of the parts of a VCR is not capable of existing separately from the VCR itself, so an ordinary VCR would not be copyrightable.

Computer programs, unlike VCRs, are copyrightable as "literary works." 17 U.S.C. § 102(a). Accordingly, one might argue, the "buttons" used to operate a computer program are not like the buttons used to operate a VCR, for they are not subject to a useful-article exception. The response, of course, is that the arrangement of buttons on a VCR would not

be copyrightable even without a useful-article exception, because the buttons are an uncopyrightable "method of operation." Similarly, the "buttons" of a computer program are also an uncopyrightable "method of operation."

That the Lotus menu command hierarchy is a "method of operation" becomes clearer when one considers program compatibility. Under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. For example, if the user wanted the computer to print material, then the user would have to learn not just one method of operating the computer such that it prints, but many different methods. We find this absurd. The fact that there may be many different ways to operate a computer program, or even many different ways to operate a computer program using a set of hierarchically arranged command terms, does not make the actual method of operation chosen copyrightable; it still functions as a method for operating the computer and as such is uncopyrightable.

Consider also that users employ the Lotus menu command hierarchy in writing macros. Under the district court's holding, if the user wrote a macro to shorten the time needed to perform a certain operation in Lotus 1-2-3, the user would be unable to use that macro to shorten the time needed to perform that same operation in another program. Rather, the

user would have to rewrite his or her macro using that other program's menu command hierarchy. This is despite the fact that the macro is clearly the user's own work product. We think that forcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in § 102(b) that "methods of operation" are not copyrightable. That programs can offer users the ability to write macros in many different ways does not change the fact that, once written, the macro allows the user to perform an operation automatically. As the Lotus menu command hierarchy serves as the basis for Lotus 1-2-3 macros, the Lotus menu command hierarchy is a "method of operation."

In holding that expression that is part of a "method of operation" cannot be copyrighted, we do not understand ourselves to go against the Supreme Court's holding in Feist. In Feist, the Court explained:

The primary objective of copyright is not to reward the labor of authors, but to promote the Progress of Science and useful Arts. To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work.

Feist, 499 U.S. at 349-50 (quotations and citations omitted). We do not think that the Court's statement that "copyright assures authors the right to their original expression" indicates that all expression is necessarily copyrightable; while original expression is necessary for copyright

protection, we do not think that it is alone sufficient. Courts must still inquire whether original expression falls within one of the categories foreclosed from copyright protection by § 102(b), such as being a "method of operation."

We also note that in most contexts, there is no need to "build" upon other people's expression, for the ideas conveyed by that expression can be conveyed by someone else without copying the first author's expression.¹³ In the context of methods of operation, however, "building" requires the use of the precise method of operation already employed; otherwise, "building" would require dismantling, too. Original developers are not the only people entitled to build on the methods of operation they create; anyone can. Thus, Borland may build on the method of operation that Lotus designed and may use the Lotus menu command hierarchy in doing so.

Our holding that methods of operation are not limited to abstractions goes against Autoskill, 994 F.2d at 1495 n.23, in which the Tenth Circuit rejected the defendant's argument that the keying procedure used in a computer program was an uncopyrightable "procedure" or "method of operation" under § 102(b). The program at issue, which was designed to test and train students with reading deficiencies, id. at 1481, required students to select responses to the program's queries "by

13. When there are a limited number of ways to express an idea, however, the expression "merges" with the idea and becomes uncopyrightable. Morrissey, 379 F.2d at 678-79.

pressing the 1, 2, or 3 keys." Id. at 1495 n.23. The Tenth Circuit held that, "for purposes of the preliminary injunction, . . . the record showed that [this] keying procedure reflected at least a minimal degree of creativity," as required by Feist for copyright protection. Id. As an initial matter, we question whether a programmer's decision to have users select a response by pressing the 1, 2, or 3 keys is original. More importantly, however, we fail to see how "a student select[ing] a response by pressing the 1, 2, or 3 keys," id., can be anything but an unprotectable method of operation.¹⁴

III.

Conclusion

Because we hold that the Lotus menu command hierarchy is uncopyrightable subject matter, we further hold that Borland did not infringe Lotus's copyright by copying it. Accordingly, we need not consider any of Borland's affirmative defenses. The judgment of the district court is

Reversed.

Concurrence follows.

14. The Ninth Circuit has also indicated in dicta that "menus, and keystrokes" may be copyrightable. Brown Bag Software v. Symantec Corp., 960 F.2d 1465, 1477 (9th Cir.), cert. denied, BB Asset Management, Inc. v. Symantec Corp., 113 S. Ct. 198 (1992). In that case, however, the plaintiff did not show that the defendant had copied the plaintiff's menus or keystrokes, so the court was not directly faced with whether the menus or keystrokes constituted an unprotectable method of operation. Id.

BOUDIN, Circuit Judge, concurring. The importance of this case, and a slightly different emphasis in my view of the underlying problem, prompt me to add a few words to the majority's tightly focused discussion.

I.

Most of the law of copyright and the "tools" of analysis have developed in the context of literary works such as novels, plays, and films. In this milieu, the principal problem-- simply stated, if difficult to resolve--is to stimulate creative expression without unduly limiting access by others to the broader themes and concepts deployed by the author. The middle of the spectrum presents close cases; but a "mistake" in providing too much protection involves a small cost: subsequent authors treating the same themes must take a few more steps away from the original expression.

The problem presented by computer programs is fundamentally different in one respect. The computer program is a means for causing something to happen; it has a mechanical utility, an instrumental role, in accomplishing the world's work. Granting protection, in other words, can have some of the consequences of patent protection in limiting other people's ability to perform a task in the most efficient manner. Utility does not bar copyright (dictionaries may be copyrighted), but it alters the calculus.

Of course, the argument for protection is undiminished, perhaps even enhanced, by utility: if we want more of an intellectual product, a temporary monopoly for the creator provides incentives for others to create other, different items in this class. But the "cost" side of the equation may be different where one places a very high value on public access to a useful innovation that may be the most efficient means of performing a given task. Thus, the argument for extending protection may be the same; but the stakes on the other side are much higher.

It is no accident that patent protection has preconditions that copyright protection does not--notably, the requirements of novelty and non-obviousness--and that patents are granted for a shorter period than copyrights. This problem of utility has sometimes manifested itself in copyright cases, such as Baker v. Selden, 101 U.S. 99 (1879), and been dealt with through various formulations that limit copyright or create limited rights to copy. But the case law and doctrine addressed to utility in copyright have been brief detours in the general march of copyright law.

Requests for the protection of computer menus present the concern with fencing off access to the commons in an acute form. A new menu may be a creative work, but over time its importance may come to reside more in the investment that has been made by users in learning the menu and in building their

own mini-programs--macros--in reliance upon the menu. Better typewriter keyboard layouts may exist, but the familiar QWERTY keyboard dominates the market because that is what everyone has learned to use. See P. David, CLIO and the Economics of QWERTY, 75 Am. Econ. Rev. 332 (1985). The QWERTY keyboard is nothing other than a menu of letters.

Thus, to assume that computer programs are just one more new means of expression, like a filmed play, may be quite wrong. The "form"--the written source code or the menu structure depicted on the screen--look hauntingly like the familiar stuff of copyright; but the "substance" probably has more to do with problems presented in patent law or, as already noted, in those rare cases where copyright law has confronted industrially useful expressions. Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit.

All of this would make no difference if Congress had squarely confronted the issue, and given explicit directions as to what should be done. The Copyright Act of 1976 took a different course. While Congress said that computer programs might be subject to copyright protection, it said this in very general terms; and, especially in § 102(b), Congress adopted a string of exclusions that if taken literally might easily seem to exclude most computer programs from protection. The only

detailed prescriptions for computers involve narrow issues (like back-up copies) of no relevance here.

Of course, one could still read the statute as a congressional command that the familiar doctrines of copyright law be taken and applied to computer programs, in cookie cutter fashion, as if the programs were novels or play scripts. Some of the cases involving computer programs embody this approach. It seems to me mistaken on two different grounds: the tradition of copyright law, and the likely intent of Congress.

The broad-brush conception of copyright protection, the time limits, and the formalities have long been prescribed by statute. But the heart of copyright doctrine--what may be protected and with what limitations and exceptions--has been developed by the courts through experience with individual cases. B. Kaplan, An Unhurried View of Copyright 40 (1967). Occasionally Congress addresses a problem in detail. For the most part the interstitial development of copyright through the courts is our tradition.

Nothing in the language or legislative history of the 1976 Act, or at least nothing brought to our attention, suggests that Congress meant the courts to abandon this case-by-case approach. Indeed, by setting up § 102(b) as a counterpoint theme, Congress has arguably recognized the tension and left it for the courts to resolve through the development of case law. And case law development is adaptive: it allows new problems

to be solved with help of earlier doctrine, but it does not preclude new doctrines to meet new situations.

II.

In this case, the raw facts are mostly, if not entirely, undisputed. Although the inferences to be drawn may be more debatable, it is very hard to see that Borland has shown any interest in the Lotus menu except as a fall-back option for those users already committed to it by prior experience or in order to run their own macros using 1-2-3 commands. At least for the amateur, accessing the Lotus menu in the Borland Quattro or Quattro Pro program takes some effort.

Put differently, it is unlikely that users who value the Lotus menu for its own sake--independent of any investment they have made themselves in learning Lotus' commands or creating macros dependent upon them--would choose the Borland program in order to secure access to the Lotus menu. Borland's success is due primarily to other features. Its rationale for deploying the Lotus menu bears the ring of truth.

Now, any use of the Lotus menu by Borland is a commercial use and deprives Lotus of a portion of its "reward," in the sense that an infringement claim if allowed would increase Lotus' profits. But this is circular reasoning: broadly speaking, every limitation on copyright or privileged use diminishes the reward of the original creator. Yet not every writing is copyrightable nor every use an infringement. The

provision of reward is one concern of copyright law, but it is not the only one. If it were, copyrights would be perpetual and there would be no exceptions.

The present case is an unattractive one for copyright protection of the menu. The menu commands (e.g., "print," "quit") are largely for standard procedures that Lotus did not invent and are common words that Lotus cannot monopolize. What is left is the particular combination and sub-grouping of commands in a pattern devised by Lotus. This arrangement may have a more appealing logic and ease of use than some other configurations; but there is a certain arbitrariness to many of the choices.

If Lotus is granted a monopoly on this pattern, users who have learned the command structure of Lotus 1-2-3 or devised their own macros are locked into Lotus, just as a typist who has learned the QWERTY keyboard would be the captive of anyone who had a monopoly on the production of such a keyboard. Apparently, for a period Lotus 1-2-3 has had such sway in the market that it has represented the de facto standard for electronic spreadsheet commands. So long as Lotus is the superior spreadsheet--either in quality or in price--there may be nothing wrong with this advantage.

But if a better spreadsheet comes along, it is hard to see why customers who have learned the Lotus menu and devised macros, for it should remain captives of Lotus because of an

investment in learning made by the users and not by Lotus. Lotus has already reaped a substantial reward for being first; assuming that the Borland program is now better, good reasons exist for freeing it to attract old Lotus customers: to enable the old customers to take advantage of a new advance, and to reward Borland in turn for making a better product. If Borland has not made a better product, then customers will remain with Lotus anyway.

Thus, for me the question is not whether Borland should prevail but on what basis. Various avenues might be traveled, but the main choices are between holding that the menu is not protectable by copyright and devising a new doctrine that Borland's use is privileged. No solution is perfect and no intermediate appellate court can make the final choice.

To call the menu a "method of operation" is, in the common use of those words, a defensible position. After all, the purpose of the menu is not to be admired as a work of literary or pictorial art. It is to transmit directions from the user to the computer, i.e., to operate the computer. The menu is also a "method" in the dictionary sense because it is a "planned way of doing something," an "order or system," and (aptly here) an "orderly or systematic arrangement, sequence or the like." Random House Webster's College Dictionary 853 (1991).

A different approach would be to say that Borland's use is privileged because, in the context already described, it is not seeking to appropriate the advances made by Lotus' menu; rather, having provided an arguably more attractive menu of its own, Borland is merely trying to give former Lotus users an option to exploit their own prior investment in learning or in macros. The difference is that such a privileged use approach would not automatically protect Borland if it had simply copied the Lotus menu (using different codes), contributed nothing of its own, and resold Lotus under the Borland label.

The closest analogue in conventional copyright is the fair use doctrine. E.g., Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539 (1985). Although invoked by Borland, it has largely been brushed aside in this case because the Supreme Court has said that it is "presumptively" unavailable where the use is a "commercial" one. See id. at 562. In my view, this is something less than a definitive answer; "presumptively" does not mean "always" and, in any event, the doctrine of fair use was created by the courts and can be adapted to new purposes.

But a privileged use doctrine would certainly involve problems of its own. It might more closely tailor the limits of copyright protection to the reasons for limiting that protection; but it would entail a host of administrative problems that would cause cost and delay, and would also reduce

the ability of the industry to predict outcomes. Indeed, to the extent that Lotus' menu is an important standard in the industry, it might be argued that any use ought to be deemed privileged.

In sum, the majority's result persuades me and its formulation is as good, if not better, than any other that occurs to me now as within the reach of courts. Some solutions (e.g., a very short copyright period for menus) are not options at all for courts but might be for Congress. In all events, the choices are important ones of policy, not linguistics, and they should be made with the underlying considerations in view.