

49 F.3d 807
United States Court of Appeals,
First Circuit.

LOTUS DEVELOPMENT CORPORATION, Plaintiff, Appellee,
v.
BORLAND INTERNATIONAL, INC., Defendant, Appellant.

No. 93-2214.

Heard Oct. 6, 1994.

Decided March 9, 1995.

Opinion

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 *810 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.

On July 31, 1992, 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 constructed using different commands and a different command structure from those of Lotus 1-2-3. In fact, Borland has constructed 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 *811 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 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 *812 macro), in which case they interpreted the macro as having been written 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 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.

**813 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, 111 S.Ct. 1282, 1296, 113 L.Ed.2d 358 (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 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*, 510 U.S. 916, 114 S.Ct. 307, 126 L.Ed.2d 254 (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, 25 L.Ed. 841 (1879). In *Baker v. Selden*, the Court held that Selden’s copyright over the textbook in which he explained *814 his new way to do accounting did not grant him a monopoly on the use

of his accounting system.⁶ Borland argues:

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 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 infringing 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 *815 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).

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 *816 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 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 an 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, 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 to operating Lotus 1–2–3, we do not inquire further whether that method of operation could have been designed differently. The “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 *817 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 *818 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, 111 S.Ct. at 1290 (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 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 *819 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.

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, 25 L.Ed. 841 (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 *820 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 be 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 or every use an infringement. The provision of reward is ***821** 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, 105 S.Ct. 2218, 85 L.Ed.2d 588 (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, 105 S.Ct. at 2231–32. *But see Campbell v. Acuff–Rose Music, Inc.*, 510 U.S. 569, —, 114 S.Ct. 1164, 1174, 127 L.Ed.2d 500 (1994). 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 on copyright protection to the reasons for limiting that protection; but it would entail a host of administrative problems that would cause *822 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.

Footnotes	
1	Judge Keeton presided over both the <i>Paperback</i> litigation and this case.

2	<p>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?).”</p> <p>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. <i>See Borland III</i>, 831 F.Supp. at 208.</p>
3	<p>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.</p>
4	<p>See <i>Borland IV</i>, 831 F.Supp. at 226–27, for a more detailed explanation of macros and the Key Reader.</p>
5	<p>Computer programs receive copyright protection as “literary works.” <i>See</i> 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, <i>disks</i>, or cards, in which they are embodied” (emphasis added)); <i>see also</i> H.R.Rep. No. 1476, 94th Cong., 2d Sess. 54 (1976), <i>reprinted in</i> 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.”).</p>
6	<p>Selden’s system of double-entry bookkeeping is the now almost-universal T-accounts system.</p>
7	<p>We consider the <i>Altai</i> test because both parties and many of the <i>amici</i> focus on it so heavily. Borland, in particular, is highly critical of the district court for not employing the <i>Altai</i> 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 <i>Altai</i> test.</p>
8	<p>We recognize that <i>Altai</i> 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, <i>Nimmer on Copyright</i> § 13.03[A](2) (1993).</p>
9	<p>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. <i>See Morrissey v. Procter & Gamble Co.</i>, 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).</p>
10	<p>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.</p>
11	<p>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. <i>See, e.g., Altai</i>, 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).</p>

12	We think that the <i>Altai</i> test would contemplate this being the initial inquiry.
13	When there are a limited number of ways to express an idea, however, the expression “merges” with the idea and becomes uncopyrightable. <i>Morrissey</i> , 379 F.2d at 678–79.
14	The Ninth Circuit has also indicated in dicta that “menus, and keystrokes” may be copyrightable. <i>Brown Bag Software v. Symantec Corp.</i> , 960 F.2d 1465, 1477 (9th Cir.), <i>cert. denied</i> , <i>BB Asset Management, Inc. v. Symantec Corp.</i> , 506 U.S. 869, 113 S.Ct. 198, 121 L.Ed.2d 141 (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. <i>Id.</i>