COPYRIGHT AND COMPUTER PROGRAMS: IT'S ALL IN THE DEFINITION

WILLIAM F. PATRY*

[*] Associate Professor of Law, Benjamin N. Cardozo School of Law, Yeshiva University, New York. Formerly counsel, Subcommittee on Intellectual Property & Judicial Administration, Committee on the Judiciary, U.S. House of Representatives; Policy Planning Advisor to the Register of Copyrights, U.S. Copyright Office. I wish to thank Professor Craig Joyce for his indefatigable efforts to give this article structural coherence, and Professors Marci Hamilton, Stewart Sterk, Jane Ginsburg, and Lee Hollaar as well as Tom Olson for generously reviewing earlier drafts. I benefitted tremendously from their comments. The staff of the *Cardozo Arts & Entertainment Law Journal*—particularly Jodi-Lynne Bayrd, Laura Eng, Daniel Goldfisher, David Goldsmith, Neil Koren, and Evan Seideman—deserve much praise for their hard work and grace under pressure.

In cases of statutory construction, we begin with the language of the statute.
Diamond v. Diehr, 450 U.S. 175, 182 (1981) (Rehnquist, J.) (patent case).

I. INTRODUCTION

In the United States, copyright has an instrumental purpose of promoting "the Progress of Science"[1] through the grant of incentives to authors. This purpose also is reflected in the works protected, not only works of the fine arts, but works of utility as well.[2] Unfortunately, some judges have regarded copyright as appropriate only for the *belle lettres*, reading into the Constitution a romantic notion of authorship.[3] An early flashpoint for prejudice against copyright in useful works is seen in nineteenth century decisions

[1] U.S. CONST. art. I, § 8, cl. 8 (Congress shall have power "[t]o promote the Progress of Science . . . by securing for limited Times to Authors . . . the exclusive Right to their respective Writings . . . .").

[2] The first Copyright Act reflected Congress's instrumental purpose by protecting books, maps, and charts; the latter two types of works were particularly useful in a young nation's efforts to expand. Act of May 31, 1790, ch. 15, 1 Stat. 124 (1790); *see* WILLIAM F. PATRY, 1 COPYRIGHT LAW & PRACTICE 25-36 (1994).

[3] For criticism of the romantic view of authorship, see Peter Jaszi, *On the Author Effect: Contemporary Copyright and Collective Creativity*, 10 CARDOZO ARTS & ENT. L.J. 293 (1992); Peter Jaszi, *Toward a Theory of Copyright: The Metamorphoses of "Authorship,"* 1991 DUKE L.J. 455.

denying protection to advertisements because of their commercial nature.[4] This attitude toward advertisements as "tuft hunters in the polite society of copyrights"[5] ended with Justice Holmes's 1903 opinion in *Bleistein v. Donaldson Lithographing Co.*[6] *Bleistein* is one of the linchpins of modern copyright law due to its establishment of an objective standard of originality; a standard by which the question of copyrightability is limited to an examination of whether sufficient creativity exists, rather than an inquiry into whether the work[7] advances the cause of "art" or "science."[8]

In the twentieth century, continuous technological innovation has led to increased corporate authorship[9] and to new forms of expression resulting from such collective endeavors, beginning with motion pictures and, fairly recently, computer programs. In drafting the 1976 Copyright Act, Congress expressed an intention to stay ahead of the technological curve by crafting language flexible enough to take into account new types of works without the need for further amendment:

> Under the bill it makes no difference what the form, manner, or medium of fixation may be—whether it is in words, numbers, notes, sounds, pictures, or any other graphic or symbolic indicia, whether embodied in a physical object in written, printed, photographic, sculptural, punched, magnetic, or any other stable form, and whether it is capable of perception directly or by means of any machine or device "now known or later

[4] *See* PATRY, *supra* note 2, at 124-25.

[5] HERBERT A. HOWELL, THE COPYRIGHT LAW 14 (3d ed. 1952). A "tuft hunter" has been defined as:

> *n.*, 1. (at Oxford or Cambridge Universities) a person who seeks the acquaintance of titled undergraduates. 2. any toady; sycophant.

WEBSTER'S NEW UNIVERSAL UNABRIDGED DICTIONARY 1523 (1992 ed.).

[6] 188 U.S. 239 (1903). *Bleistein* did not, however, herald an unbroken line of cases reflecting a more modern approach to copyright. Five years later, in White-Smith Music Publishing Co. v. Apollo Co., 209 U.S. 1 (1908), the Court held that a perforated piano roll was not a "copy" of a musical composition reproduced therein because it was not readable by the consumer. The following year, Congress imposed liability under these circumstances (subject to the world's first compulsory license), but it was not until the 1976 Copyright Act that Congress reversed *White-Smith*'s holding on "copy." *See* PATRY, *supra* note 2, at 518, 573.

[7] It is the system as a whole, rather than any particular work, which must "promote the Progress of Science." U.S. CONST. art. I, § 8, cl. 8. Accordingly, courts have upheld protection in pornographic works. *See* PATRY, *supra* note 2, at 126-27.

[8] *See* Weissman v. Freeman, 868 F.2d 1313 (2d Cir.), *cert. denied*, 493 U.S. 883 (1989). The district court had denied protection to a revision of a scientific article because, in its opinion, the revision did not contain sufficient scientific advances. *See* Weissman v. Freeman, 684 F. Supp. 1248, 1261 (S.D.N.Y. 1988).

[9] *See* Marci A. Hamilton, *Appropriation Art and the Imminent Decline in Authorial Control over Copyrighted Works*, 42 J. COPYRIGHT SOC'Y 93, 99, 105-07 (1994); Marci A. Hamilton, Note, *Commissioned Works as Works Made for Hire Under the 1976 Copyright Act: Misinterpretation and Injustice*, 135 U. PA. L. REV. 1281, 1319 (1987).

developed."[10]

Old habits die hard, however. In the last ten years, computers, in addition to revolutionizing our lives, have presented the courts with a number of difficult issues regarding copyrightability and infringement. Like their 19th century predecessors, a number of judges have been antagonistic to Congress's decision to protect computer programs under the copyright law.[11] Even those courts favorably disposed towards such protection have resorted inappropriately to analogies to other subject matter, such as plays, in an attempt to fit computer programs into familiar models.[12]

Three cases decided within the past decade illustrate the difficulties courts have encountered (or created for themselves): *Whelan Associates v. Jaslow Dental Laboratory, Inc.*,[13] *Computer Associates International, Inc. v. Altai, Inc.*,[14] and *Lotus Development Corp. v. Borland International, Inc.*[15] In *Whelan Associates*, the Third Circuit, faced with a compelling case of infringement (a former business associate's unauthorized reproduction of substantial nonliteral elements), overreacted, extending protection to virtually all components of the program. In *Computer Associates*, the Second Circuit, perhaps in response to *Whelan Associates*, swung the pendulum too far the other direction, excluding from protection the essence of a computer program: its flow, the manner in which the program's set of statements or instructions are combined to bring about a "certain result." In *Lotus Development Corp.*, the First Circuit denied all protection to a highly creative component of Lotus's popular 1-2-3 electronic spreadsheet program (its menu command hierarchy), on the ground that the hierarchy constituted a "method of operation."

All three courts failed to determine the copyrightability of computer programs by reference to the basic touchstone of American copyright law, originality. In addition to overlooking the lessons of *Bleistein*, the courts also overlooked a critical feature of the 1976 Copyright Act,[16] section 101, which contains the Act's principal definitions.[17] The *Computer Associates* and *Lotus Development*

[10] H.R. Rep. No. 1476, 94th Cong., 2d Sess. 52 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5665-66; S. Rep. No. 473, 94th Cong., 1st Sess. 51 (1975).

[11] *See infra* note 23 and accompanying text.

[12] *See infra* text accompanying notes 91-97.

[13] 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

[14] 982 F.2d 693 (2d Cir. 1992).

[15] 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996).

[16] Copyright Act of 1976, 90 Stat. 2544 (1976) (codified as amended at 17 U.S.C. §§ 101-803 (1994), 18 U.S.C. § 2319 (1994)).

[17] 17 U.S.C. § 101. In instances involving compulsory licensing of particular industries,

Corp. courts, moreover, had an authoritative guide to the significance of the definitions. In *Feist Publications, Inc. v. Rural Telephone Service Co.*,[18] the Supreme Court, per Justice O'Connor, regarded these definitions not as mere adjuncts to subsequent sections of the Act, but instead as central to the statutory scheme: the definitions embody Congress's judgment about how the requirement of originality is to be satisfied for the defined subject matter.[19] In *Atari Games Corp. v. Oman*,[20] Judge (now Justice) Ruth Bader Ginsburg similarly regarded the definition of "audiovisual works," rebuking the Copyright Office for its repeated failure to recognize that copyrightability was to be determined by construing "the definitional term."[21]

By contrast, the *Lotus Development Corp.*, *Computer Associates*, and *Whelan Associates* courts determined the copyrightability of computer programs without construing the statutory definition of "computer program." Indeed, the First Circuit in *Lotus Development Corp.*[22] did not even bother to cite the definition. Based on a mistaken belief that Congress has not given them directions in the statute,[23] the lower courts have usurped the role exclusively reserved in the Constitution for Congress[24] by crafting their own solutions to the scope of computer program protection. The lower courts' solutions are enunciated in inordinately complicated tests

Congress provided specialized definitions. *See* 17 U.S.C. §§ 111(f) (cable television), 118(g) (public broadcasting), 119(d) (satellite transmissions).

[18] 499 U.S. 340 (1991).

[19] *Id.* at 356 (construing definition of "compilation" as constituting a three-part test of originality).

[20] 888 F.2d 878 (D.C. Cir. 1989).

[21] *Atari Games Corp.*, 888 F.2d at 883; *accord* Atari Games Corp. v. Oman, 979 F.2d 242, 244-45 (D.C. Cir. 1992). The District of Columbia Circuit remanded this case to the Copyright Office, and subsequently reversed the Copyright Office's decision. *Atari Games Corp.*, 979 F.2d 242. In *Computer Associates*, the Second Circuit cited the definition in passing, but did not analyze or apply it. *Computer Assocs.*, 982 F.2d at 697.

[22] 49 F.3d 807.

[23] *See, e.g.*, *Lotus Dev. Corp.*, 49 F.3d at 820 ("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.") (Boudin, J., concurring).

Undoubtedly, the authors of the *Computer Associates* and *Lotus Development Corp.* opinions would argue in good faith that they *were* following the course prescribed for them by Congress: to apply familiar doctrines of copyright law in the absence of any indication to the contrary. *See Computer Assocs.*, 982 F.2d at 712 ("If the test we have outlined results in narrowing the scope of protection, as we expect it will, that result flows from applying, in accordance with Congressional intent, long-standing principles of copyright law to computer programs."). The definition of "computer program" in section 101 of the Copyright Act does provide to the contrary, though, while the judges' open second-guessing of the wisdom of Congress's decision to protect computer programs as copyrighted works, *see id.*, suggests something more than mere fealty to Congress's wishes.

[24] *See supra* note 1.

with imposing names such as "abstraction-filtration-comparison,"[25] that ruthlessly dissect and boil down computer programs into a faint residue of their formerly robust selves.[26] Rather than give substance to the whole of the work as Congress intended, these tests direct the district courts to search for "golden nuggets" much like pigs hunt for truffles. Ignoring Congress's judgment and the objective standard of originality, the lower courts have instead determined the copyrightability of computer programs by using subjective and evolving[27] concepts such as efficiency[28] and compatibility.[29] The result has been a deliberate exclusion from protection of material that properly falls within the scope of copyright.

The First Circuit's *Lotus Development Corp.* opinion provides an extreme example.[30] As support for its conclusion that Lotus's 1-2-3 menu command hierarchy[31] is a "method of operation" nonprotectible under section 102(b) of the Act, the court of appeals wrote:

> 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.[32]

[25] *See Computer Assocs.*, 982 F.2d at 706-12.

[26] *See* Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright Over Sui Generis Protection for Computer Software*, 94 COLUM. L. REV. 2559, 2561 (1994) (criticizing these tests for " 'dissect[ing]' the work [so] as to classify all its elements as unprotectable."); *see also infra* text accompanying notes 223-43.

[27] I use "evolving" in contradistinction to "static." The concept of efficiency is evolving because new products are constantly being created that represent a significant advance over earlier works. Each new product may be regarded, at the time of its introduction, as "the most efficient," that is, until the next product comes along with an even more "efficient" solution. In light of these ever-shifting benchmarks, it is inappropriate to determine legal rights by reference to evolving concepts such as efficiency as the Second Circuit did in *Computer Associates.* Interestingly, in 1992, Borland described the Lotus 1-2-3 menu command hierarchy as an "outdated user interface." *See* Brief for Petitioner at 16, Lotus Dev. Corp. v. Borland Int'l, Inc., 116 S. Ct. 804 (1996) (No. 94-2003).

[28] *See infra* text accompanying notes 238-43.

[29] *See infra* text accompanying notes 32-34 & note 34.

[30] *Lotus Dev. Corp.*, 49 F.3d 807; *see infra* text accompanying notes 46-78, 244-69.

[31] *See infra* text accompanying notes 46-78 (further discussion of the nature of the 1-2-3 menu command hierarchy).

[32] *Lotus Dev. Corp.*, 49 F.3d at 817-18.

> 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.

Id. at 818.

Leaving aside the absence of any statutory reference to a compatibility limitation on protection,[33] the inaccurate use of the term "compatibility,"[34] and the tortured reading of section 102(b),[35] it is apparent that the court of appeals let concerns about the marketing of programs influence the very different legal question of whether the Lotus 1-2-3 menu command hierarchy is an "original work[ ] of authorship."[36] Under the First Circuit's approach it does not matter how original Lotus's work is, nor whether there are numerous other spreadsheet programs, each with non-"substantially similar" original menu command hierarchies and that compete with Lotus's. The critical issue according to the First Circuit is not the content of the works, but the possibility that a user of 1-2-3 may wish to switch to one of Lotus's competitors. In order to make switching as attractive as possible, at least as long as Lotus has

[33] Other courts have rejected compatibility defenses. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

> [Defendant] may wish to achieve total compatibility with independently developed application programs written for [plaintiff's work], but that is a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged.

Id. at 1253.

[34] There is no one meaning for the term "compatibility." The term is properly used broadly to refer to the complementary operation of computers, computer peripherals, or computer programs. Complementary activities include having two computers manufactured by different companies communicate with each other, for example, having a printer manufactured by one company (such as Hewlett Packard) operate with a different company's computer (such as IBM).

The First Circuit, however, used "compatibility" in a very different sense: copying by competitors seeking to sell a substitute product. Competitive activities include copying portions of copyrighted operating system computer programs in order to permit existing applications programs written for one machine (such as an Apple) to run on a clone manufacturer's computer (such as a Franklin) so that consumers will have an incentive to purchase the clone rather than the original, and, as in *Lotus Development Corp.*, the copying of files from an applications computer program (such as Lotus 1-2-3) into a competing applications program (such as Quattro Pro) so that the 1-2-3 files will "run" on Quattro Pro, thereby encouraging consumers to purchase Quattro Pro.

To use "compatibility" in connection with Quattro Pro, as the First Circuit did, is misleading because it may infer there was a technical reason for Borland's copying of Lotus's 1-2-3 menu command hierarchy. There wasn't. The reason Borland copied the menu command hierarchy was to lure Lotus's customers away by permitting them to transport their "macros" to Quattro Pro. *See infra* text accompanying notes 71-78. While this was an understandable business objective, it was not motivated by technical concerns. Indeed, Borland believed that Quattro Pro was superior to 1-2-3. *See* Brief for Petitioner, *Lotus Dev. Corp.*, *supra* note 27, at 13.

[35] 17 U.S.C. § 102(b). The First Circuit's misapplication of § 102(b) is discussed *infra* text accompanying notes 213-14.

[36] 17 U.S.C. § 102(a). The First Circuit's concession that the menu command hierarchy contained expressive elements, *see Lotus Dev. Corp.*, 49 F.3d at 816, is at least semantically inconsistent with its finding of no protection. "Expressive elements" is the very stuff of protection. *Cf.* Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78, 90 (D. Mass. 1992) ("It is not entirely clear whether the argument is that anything 'functional' is not 'expressive,' or instead that even 'expressive' elements of anything that is 'functional' are not 'copyrightable.' The choice between these two different ways of phrasing the argument seems, in any event, more linguistic than substantive.").

an important[37] market share, under the First Circuit's approach each of Lotus's competitors must be permitted to copy the entirety of Lotus's menu commands.[38]

This article proposes that, unlike the lower courts did, one must begin with the language and structure of the statute in addressing the copyrightability and infringement of computer programs. The pertinent language is found in section 102(a)'s extension of protection to "original works of authorship" and in the section 101 definition of "computer program."[39] The pertinent structure of the Copyright Act is the interrelationship between these two sections. Section 102(a) is the source of all protection under the Act. Incorporating the constitutional requirement of originality, the section limits protection to those works that are independently created and possess "at least some

[37] The marketing history of personal computer software, where dominance is usually short-lived, presents compelling evidence of the inadvisability of making legal judgments on copyrightability turn on transitory business factors. For example, it cannot be said that Lotus 1-2-3 has a dominant market share. Microsoft's "Excel," which performs the same functions as 1-2-3 and in a noninfringing manner, generally is considered the current industry leader. Lotus has been sold to IBM, which purchased the company not because of 1-2-3 but because of Lotus's popular "Notes" program. Borland previously had been sold to Novell for $125 million. A recent article in the *Wall Street Journal* indicates that Novell is now attempting to sell Quattro Pro at a fire-sale price after disappointing returns on its investment. *See* Don Clark, *Novell in Talks to Sell Software Lines it Acquired in WordPerfect Purchase*, Wall St. J., Oct. 31, 1995, at A3.

In Douglas G. Baird et al., Game Theory and the Law 212-13 (1994), it is argued that the 1-2-3 menu commands are a "network externality" in economic terms, and as such should not be protected. Otherwise existing users of 1-2-3 will allegedly not purchase superior, competing programs. The existence of this installed 1-2-3 customer base justifies, the authors assert, Borland's copying of the entire 1-2-3 command hierarchy. This approach to copyright turns the Constitution on its head; successful, creative works will be denied protection solely because of their success, while mediocre works will receive more protection because of their mediocrity.

A different economic approach is Joseph Schumpeter's theory of "creative destruction," developed in the nineteenth century. This theory proposes that requiring later-comers to invent around the first work will force competitors to develop products that leapfrog ahead and demonstrate sufficient superiority to make them attractive to consumers using an existing product. *See* Erich W. Streissler, *The Influence of German and Austrian Economics on Joseph A. Schumpeter, reprinted in* Schumpeter in the History of Ideas 19-22 (Yuichi Shiomaya & Mark Perlman eds., 1993). The market evidence favors Schumpeter over Baird et al. in the case of electronic spreadsheets. As noted above, Microsoft's Excel spreadsheet managed to become the leader in the spreadsheet market, pulling customers away from Lotus and Borland and attracting new customers. Microsoft did not need to copy, and did not copy, Lotus's 1-2-3 menu command hierarchy to accomplish this. If Lotus 1-2-3 had been denied protection for its menu command hierarchy, Microsoft would not have been required to engage in "creative destruction;" it simply could have copied from Lotus and added a few new features. The end result would have been a less inventive product. It is far from clear, therefore, that denying protection to a dominant product will always better serve consumers; the opposite may be the case.

[38] For a discussion of the technical reasons behind the court of appeals's compatibility comment, see *infra* text accompanying notes 76-78.

[39] 17 U.S.C. § 101, quoted *infra* text accompanying note 139.

minimal degree of creativity."[40] Section 102(a) does not inform us, however, how this threshold requirement of originality can be met for any particular class of works. The answer to that question is provided by the section 101 subject matter definitions,[41] including the definition of "computer program."

The suggested approach—to regard the section 101 subject matter definitions as embodying Congress's expression of how originality can be satisfied—originates with Judge (now Justice) Ruth Bader Ginsburg's opinion in *Atari Games Corp. v. Oman*,[42] and with Justice O'Connor's opinion for the Court in *Feist Publications, Inc. v. Rural Telephone Service Co.*[43]

Lotus Development Corp. v. Borland International, Inc., for which the Court granted certiorari on September 27, 1995,[44] provided the Supreme Court with the opportunity to apply this existing methodology to issues of the copyrightability of computer programs. Unfortunately, on January 16, 1996, one week after oral argument, an equally divided Court affirmed due to Justice Ste-

[40] *Feist*, 499 U.S. at 345; *see infra* text accompanying notes 80-81.

[41] In the case of musical works, dramatic works, and pantomimes and choreographic works, Congress did not provide a definition of the subject matter, believing these works had fairly settled meanings. H.R. Rep. No. 1476, *supra* note 10, at 53, *reprinted in* 1976 U.S.C.C.A.N. at 5666-67; S. Rep. No. 473, *supra* note 10, at 52. The fact that Congress took the trouble to provide definitions for the remaining five categories of works because their meanings were unsettled should heighten the attention courts pay to those definitions.

[42] 888 F.2d at 881-82; *see infra* text accompanying notes 168-78.

[43] 499 U.S. 340; *see infra* text accompanying notes 167-78.

[44] 116 S. Ct. 39 (1995). The question presented for certiorari was:

> Whether a computer program's particular menu command hierarchy, which the district court found to contain expression separable from its underlying idea and the functionality it describes, may be protected by copyright in light of the explicit Congressional extension of copyright to computer programs under the same principles applicable to other literary works; or whether, as the First Circuit held, Section 102(b) of the Copyright Act bars protection for any such menu command hierarchy despite its expressive characteristics, because it assists users in communicating with a computer program in order to perform useful operations.

Brief for Petitioner, *Lotus Dev. Corp.*, *supra* note 27, at i.

The first question (up to the semi-colon) erroneously states that there is an "explicit congressional extension of copyright to computer programs under the same principles applicable to other literary works." *See id.* at 31 ("Congress specifically intended the expression in computer programs to enjoy copyright protection under the same principles governing other forms of original expression in literary works."). The legislative history, reviewed *infra* text accompanying notes 104-19, does not bear out this assertion. But, legislative history aside, the existence of the definition of "computer program" belies the assertion that Congress intended computer programs to be regarded like plays.

By contrast, in its brief on the merits, Lotus described the issue before the Supreme Court as "whether the copyright protection Congress granted computer programs protects the original, creative expression contained in [Lotus's] user interfaces against wholesale appropriation by those who determine to achieve commercial success through imitation rather than innovation." Brief for Petitioner, *Lotus Dev. Corp.*, *supra* note 27, at 16. This hyperbole inaccurately shifts attention to defendant's conduct in copying, rather than focusing, as should be the case, on whether what the defendant copied is protectible.

vens's recusal.[45] The important issues raised in *Lotus Development Corp.* thus remain unsolved. Before this section 101 application is spelled out, though, it is first necessary to review some of the unique facts of *Lotus Development Corp.* (part II). We will then discuss the structure of the Copyright Act, focusing on the relationship between section 102(a) and the section 101 subject matter definitions (part III). Part IV reviews the historical treatment of computer programs as literary works. Part V analyzes the idea-expression dichotomy. Part VI sets forth a suggested approach to determining the copyrightability of computer programs. Part VII traces some of the false paths the courts have taken in their attempt to determine the copyrightability of computer programs when less than the entirety of the program has been copied. Part VIII discusses the disarray in the lower courts, including *Lotus Development Corp. v. Borland International, Inc.* Part IX applies the test for protection of computer programs developed in part VI to the Lotus 1-2-3 menu command hierarchy. The article concludes with a review of safety-valve considerations the courts may take into account to ensure the proper balance is maintained between providing sufficient incentives for authors and encouraging innovation by subsequent programmers.

II. The 1-2-3 Menu Command Hierarchy

A. *A Technical Description*

At issue in *Lotus Development Corp.* is the copyrightability of the "menu command hierarchy" for Lotus's popular 1-2-3 electronic spreadsheet program. Lotus 1-2-3 is a powerful, multi-purpose computer program that integrates three functions: (1) an electronic spreadsheet for manipulating numerical, statistical, and financial information provided by the user; (2) graphics capabilities for displaying information in charts or graphs; and (3) a database management system for storing information.

These functions are performed on a two-dimensional grid of columns and rows that frame a middle, worksheet area. On the left side of the grid is a column of numbers, beginning at the top with the number 1 and continuing down to the number 20. Across the top of the grid, beginning at the far left, are eight letters beginning with the letter A and ending with the letter H.[46] On the right side of the grid is a row of command symbols or icons, called the "icon

[45] 116 S. Ct. 804 (1996).

[46] The intersection of a letter and a number is called a "cell."

palette." Completing the frame, across the bottom, is the "status line," which gives the date, time, and any messages.

Since most computer monitors have fewer than thirteen inches of usable screen space, only a limited amount of information can be displayed at one time. For this reason, only eight columns across the top and twenty rows on the left side can be displayed to the consumer. The 1-2-3 program actually contains 8,192 rows and 256 columns, equalling 2 million cells.[47] If Lotus 1-2-3 were fixed on paper, it would be 130 feet long and twenty-one feet wide. Clearly, it is a substantial work. Lotus 1-2-3's benefits transcend its vastness and the speed with which it operates; 1-2-3 not only stores and displays information, but also calculates, analyzes, and projects complex figures. One of the principal benefits of all electronic spreadsheets (beginning with 1-2-3's predecessor "VisiCalc") is that the change of any one number on the spreadsheet causes all other related numbers to be recalculated automatically.

Consumers operate Lotus 1-2-3 through a computer keyboard (or a mouse) and by various commands[48] contained in the program that execute particular functions.[49] The commands, called "menu commands,"[50] are arranged in "menus"[51] or submenus, the entirety of which is called variously the "command structure," the "menu command structure," the "menu hierarchy," or the "menu command hierarchy."[52] Such a hierarchy or structure is necessary

[47] *See supra* note 45.

[48] *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203, 206 (D. Mass. 1992) (" 'Command' refers to an abbreviated description of a direction that a user of a software program . . . may invoke to cause some operation to be performed.").

[49] Not all of the terms contained in the menu command hierarchy perform functions. Some "describe categories of commands, and serve to guide the user to another menu, or sub-menu, of increasingly specific command choices, and so on until all the instructions necessary to cause the program to perform a particular task have been described." Brief for Petitioner, *Lotus Dev. Corp., supra* note 27, at 8. Nor are the menu commands the only commands used to perform 1-2-3 functions. For example, the menu commands do not perform arithmetical or mathematical operations. *See id.* at 8 n.12.

[50] *Id.*

> "Menu command" refers to a command that appears in a menu. In Lotus 1-2-3, a menu command is ordinarily a single English-language word. In rare instances, it is instead a´representation of an English-language pronunciation (such as "Xtract"). Menu commands are displayed on the computer monitor by the 1-2-3 program in a succession of menus. The menus communicate to the user, in sequence, the spreadsheet operations available to the user.

Lotus Dev. Corp., 799 F. Supp. at 206. *See supra* note 48 (definition of "command").

[51] *Lotus Dev. Corp.*, 799 F. Supp. at 206 (" 'Menu' refers to a display on the computer monitor of a limited number of commands available to the user at a given moment.").

[52] *Id.* (" 'Command structure' refers to the organization of the menus and menu commands. (Other phrases used with essentially the same meaning include 'menu command structure,' 'menu hierarchy,' and 'menu command hierarchy.')").

In Lotus 1-2-3, menu commands are organized so that less than a dozen related menu commands are displayed at any given moment. This display communicates to the user the spreadsheet operations immediately available. Each menu of less than a dozen commands due to the large number of commands, of which only a limited number can be displayed on a computer monitor at any one time. The 1-2-3 menu command hierarchy consists of 469 computer commands arranged into fifty menus and submenus. The creation and arrangement of the Lotus 1-2-3 menu command hierarchy required considerable ingenuity. As one independent guide to the program states: "[a]ccessing 1-2-3's commands is like travelling with Alice down the rabbit hole. Suddenly a vast wonderland of commands is available."[53] Finally, as we shall discuss shortly, the menu command hierarchy also permits the user to create "macros," instructions to the computer that automatically execute repetitive functions.[54]

B. *The Lower Courts' Rulings on the Menu Command Hierarchy*

The Borland Corporation, a competitor of Lotus, released its "Quattro" electronic spreadsheet program in November 1987, almost five years after the introduction of 1-2-3. Quattro, and its 1989 successor "Quattro Pro," copied the entirety of the Lotus 1-2-3 menu command hierarchy.[55] Lotus sued four days after a

is linked to preceding/succeeding menus by the operation of menu commands. All command menus are ultimately linked to a single main (root/trunk) menu to form a "menu tree."

[53] CHRIS GILBERT & LAURIE WILLIAMS, THE ABC'S OF 1-2-3 RELEASE 2.4 FOR DOS USERS 42 (1992); *see* Peter Huber, *Madonna Ain't Software*, FORBES, Sept. 3, 1990, at 104 ("unlike the alphabet, command structures are not a universal and ancient heritage of the spreadsheet business; they are what distinguish the good, the bad and the ugly. . . . Some, like the Lotus spreadsheet, are wonderfully transparent and intuitive.").

[54] *See infra* text accompanying notes 73-78.

[55] Borland's Quattro and Quattro Pro had different ways of using the 1-2-3 menu command hierarchy. The first version, in Quattro and in Quattro Pro versions 1.0, 2.0, 3.0, and 4.0 had both a "native" menu tree and a Lotus 1-2-3 "emulation interface." The native menu tree was an original creation of Borland and did not, apparently, contain any of Lotus's material. Lotus did not allege that Borland's native menu tree (which was substantially larger than Lotus's) was infringing. Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 202, 211 (D. Mass. 1993). The entire Lotus 1-2-3 menu command hierarchy was, however, contained in the Borland emulation interface. As Judge Keeton described the matter:

> Invoking a Borland command that is identical to a Lotus command produces a menu that is an identical copy of the Lotus submenu, but (in some cases) with one or two new commands inserted. Thus, each menu or submenu in Lotus 1-2-3 is reproduced identically, but with the insertion of the Quattro and Quattro Pro menu trees of some new commands and any submenus associated with the new menu commands. . . . Put another way, both the Quattro and Quattro Pro version 1.0 menu trees consist of a *virtually identical* copy of the entire 1-2-3 menu tree, with new branches or leaves inserted at various places.

Id. at 212.

The emulation interface must be distinguished from Borland's "Key Reader." The Key Reader was not part of the emulation interface, but rather was a substitute for it. The Key Reader contained all of the 1-2-3 menu commands in a simplified version that reproduced only the first letter of each individual command, but with a "virtually identical copy of the Lotus menu tree that Borland used for its emulation interface . . . ." *Lotus Dev. Corp.*, favorable ruling against an earlier copyist.[56]

In *Lotus Development Corp.*, Judge Keeton, who also had heard the earlier case, again held the 1-2-3 menu command hierarchy copyrightable and infringed.[57] The First Circuit reversed, finding the menu command hierarchy to be a "method of operation" within the meaning of section 102(b) of the Copyright Act. The court of appeals did not, therefore, reach the issue of infringement or any affirmative defenses such as fair use.[58] The court of appeals's analysis proceeded from two false premises, one of fact and one of law.

The erroneous factual premise is that the 1-2-3 menu command hierarchy was, as a whole, essential to the operation of the larger 1-2-3 program. Under this approach, all 469 individual commands, divided into fifty menus and submenus, became "essential" to the program's operation precisely and only because without them, the program would not run. But, this also would be true of any menu commands chosen, and regardless of how much creativity went into their selection or arrangement. The obvious (and incorrect) result of the First Circuit's decision is that all menu command structures are per se unprotected.

The erroneous legal premise is that section 102(b) is intended to eliminate from protection work containing expressive elements because the work is functional and has achieved a measure of market success such that competitors wish to copy it.

Instead of focusing initially on section 102(b), as the First Circuit did, the proper approach is to use the methodology provided by Congress and by the Supreme Court in *Feist*[59] and the Court of Appeals for the District of Columbia Circuit in *Atari Games Corp. v. Oman.*[60] One determines whether the program's "set of statements or instructions," as defined in section 101, is independently created and possesses the requisite spark of creativity to be an "original work of authorship" under section 102(a).[61]

831 F. Supp. at 229. Borland referred to the Key Reader as "phantom menus" of the 1-2-3 menu command hierarchy. *Id.* Both the emulation interface and the Key Reader were located in a Quattro Pro file labelled, appropriately, "123.mu."

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

[57] Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (denying cross motions for summary judgment), 799 F. Supp. 203, 211-16 (denying Borland's motion for summary judgment and granting, in part, Lotus's motion for summary judgment), 831 F. Supp. 202, 831 F. Supp. 223 (D. Mass. 1993).

[58] *See Lotus Dev. Corp.*, 49 F.3d 807.

[59] 499 U.S. 340; *see infra* text accompanying notes 160-78.

[60] 888 F.2d at 881-82, 979 F.2d 242. See *supra* note 21 for an explanation of case history.

[61] For a more detailed discussion of this approach, see *infra* text accompanying notes 160-78.

The vast majority of works will pass the test easily, but this does not mean that all uses of material from a protected computer program will be infringing. If the amount of creativity is small, resulting in a "thin" copyright, close copying may be required for infringement. Even for highly creative works, section 102(b) permits the use of unoriginal component parts.[62] Analyzed this way, the historic balance of the copyright system—providing adequate incentives for authors, while permitting subsequent authors to build on the work of predecessors—can be maintained. That balance is destroyed, however, by bright-line approaches such as those found in the lower courts to date.

C. *Classification of Authorship in the Lotus 1-2-3 Menu Command Hierarchy*

Lotus did not allege that Borland copied any of its computer code. Borland conceded that it copied all of the words and the arrangement of Lotus's menu command hierarchy, and did not contest the protectibility of 1-2-3 as a whole. The issue, therefore, is whether the menu command hierarchy is a copyrightable component of the larger 1-2-3 program. This task is complicated by an important, though unresolved and rarely articulated problem: the *uncertain classification* of authorship in the menu command hierarchy. Is the hierarchy a functional compilation residing in a computer? Is it a subset of the set of statements or instructions that make up the larger 1-2-3 computer program and thus should be analyzed as a computer program, or is the menu command hierarchy incapable of being classified into any existing category of works of authorship?

The First Circuit avoided deciding what type of work was involved, merely noting that the case did not involve the copying of code.[63] Yet, the First Circuit's reasoning can be read as applying to

[62] Additionally, the fair use privilege, statutorily recognized in 17 U.S.C. § 107, as well as the special limitation on computer programs in § 117, permit copying of expression. *See infra* text accompanying notes 270-72 ("safety-valve" considerations).

[63] *Lotus Dev. Corp.*, 49 F.3d at 814-15. Petitioner Lotus was also vague about the nature of the 1-2-3 menu command hierarchy. In some passages, Lotus appears to regard the hierarchy as a computer program, or at least as part of a computer program. *See* Brief for Petitioner, *Lotus Dev. Corp., supra* note 27, at 16, 18, 28-32, 45-49. At other times, however, Lotus appears to regard the hierarchy as a "user interface" to be analyzed separately from the 1-2-3 code. *See id.* at 11, 17. Because Lotus's complaint concerns the reproduction of the menu commands—few of which are visible to the user at any given time—and not their layout, *id.* at 6 n.8, and because that layout is devoid of substantial graphic elements and does not constitute a series of related images, the menu commands should not be analyzed as a pictorial or graphic work or as an audiovisual work.

Mitel v. Iqtel, Inc., 896 F. Supp. 1050 (D. Colo. 1995), at first blush raises similar issues. In *Mitel*, the plaintiff used "command codes" in its "call controller" devices, devices code as well,[64] and it is not necessary that code be involved for the work to be considered a "computer program."[65] The issue of classification is important since the manner in which the originality requirement is satisfied varies with the type of work.

1. The 1-2-3 Menu Command Hierarchy as a Programming Language

Perhaps the First Circuit avoided analyzing the nature of authorship of the menu command hierarchy because it was of the opinion that the hierarchy was a "method of operation" in the form of a "programming language," with the individual commands acting as words and the hierarchy acting as syntax. One difficulty with the argument that the menu commands are a programming language is that a number of the menu commands do not "relate to a specific operation that the program will perform. Rather, [menu commands] describe categories of commands, and serve to

which perform functions connected with long distance telephone dialing. The command codes are three or four arbitrarily chosen digits or letters entered by technicians using a telephone keypad or laptop computer. The codes were distributed in manuals and pocket reference cards rather than in a computer program. The court held the codes unprotectible under § 102(b). The fact that the codes were short, arbitrary, and not embodied in a computer program, however, distinguishes *Mitel* from *Lotus Development Corp.*

[64] In its brief in support of Lotus's petition for certiorari, the Intellectual Property Owners trade association argues:

> The First Circuit's attempt to limit its holding by distinguishing "computer code" from the Lotus command hierarchy is based on flawed and unpersuasive reasoning. The court stated that the menu command hierarchy of Lotus 1-2-3 was a method of operation because "[w]ithout 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." 49 F.3d at 815 (emphasis supplied). On the other hand, the code of Lotus 1-2-3 is not a method of operation because, in order for a competing program "to offer *the same capabilities* as Lotus 1-2-3 [a competitor does not] have to copy Lotus's underlying code. . . ." *Id.* at 816 (emphasis supplied). The court asked a different question regarding the menu command hierarchy (whether it was needed to operate *a particular program*—Lotus 1-2-3) than it asked regarding code (whether the particular code was needed to offer *the same capabilities* as Lotus 1-2-3). Had it applied to the user interface the same standard it applied to the code, the Court of Appeals would have found the menu command hierarchy copyrightable, since a program offering *the same capabilities* as Lotus 1-2-3 could have a different menu command hierarchy, as the district court found. By the same token, had it applied to the program code the same standard it applied to the user interface, it would have recognized the flaw in its reasoning. The code of Lotus 1-2-3 is necessary to operate Lotus 1-2-3, as the code of any program is necessary to operate that particular program, making the code—under the First Circuit's reasoning—an unprotectible method of operation.

Motion for Leave to File Brief Amicus Curiae and Brief Amicus Curiae of Intellectual Property Owners in Support of Petitioner at 14 n.16, Lotus Dev. Corp. v. Borland Int'l, Inc., 116 S. Ct. 804 (1996) (No. 94-2003); *see infra* text accompanying note 260.

[65] The § 101 definition of "computer program" refers to a "set of statements or instructions," not to code, and refers to the set being used directly *or* indirectly in the computer. *See infra* text accompanying notes 133-34 (legislative history of the term "indirectly").

guide the user to another menu . . . ."[66] Another problem with the argument is that, unlike accepted programming languages such as BASIC, Pascal, and C, which may be used to compose a wide variety of computer programs and are therefore of general application, the Lotus 1-2-3 menu commands may be used only within parameters set by 1-2-3. It thus seems a bit of a stretch to consider as a "language" material that can be used to communicate only within the parameters of a single work.

Judge Keeton, in *Lotus Development Corp. v. Paperback Software International*,[67] rejected the defendant's argument that the Lotus 1-2-3 menu command hierarchy was a language. He noted that there was no precedent holding that coined languages are unprotectible, and he disagreed that

> "language" and "sets of statements or instructions" are opposites, and never the twain shall meet—that is, anything that is a "set of statements or instructions" is not a "programming language," and anything that is a "programming language" is not "a set of instructions."[68]

If one adopts Judge Keeton's reasoning, the argument that the 1-2-3 menu command hierarchy is a programming language does not advance the issue of its copyrightability very far, if at all.

2. The 1-2-3 Menu Command Hierarchy as a Compilation

A more plausible view is that the Lotus 1-2-3 menu command hierarchy is a compilation of 469 terms, arranged and coordinated into fifty menus and submenus but separate from the code with which it operates. If one adopts this view, *Lotus Development Corp.* was not a "computer program" case.[69] Instead, the case involved

[66] Brief for Petitioner, *Lotus Dev. Corp.*, *supra* note 27, at 8.

[67] 740 F. Supp. 37.

[68] *Id.* at 72. For an article critical of Judge Keeton's ruling on this point, see Richard H. Stern, *Copyright in Computer Programming Languages*, 17 RUTGERS COMPUTER & TECH. L.J. 321 (1991). Mr. Stern's approach, though, is backwards, since he argues "[t]hat the court in the *Paperback* case protected a computer programming language under the copyright laws is shown by analyzing what acts of the defendants the court condemned." *Id.* at 328. A work does not become copyrightable or noncopyrightable simply because it has been copied; it becomes copyrightable by being an original work of authorship. Elsewhere, Stern acknowledges that the 1-2-3 code was written in a preexisting programming language of general application, and that the menu command hierarchy originated with Lotus. *Id.* at 357. He also does not argue that the menu command hierarchy lacks the requisite spark of creativity. Ultimately, then, Stern's argument is the same as the First Circuit's: the menu command hierarchy, although original to Lotus, is unprotectible because it is functional; in the words of the definition of "computer program" in § 101, it "brings about a certain result." No doubt, this is why Judge Keeton rejected Paperback's identical argument as a mere "word game." *Paperback*, 740 F. Supp. at 72.

[69] Although not explicitly holding that the menu command hierarchy was a compila-

the next level case after *Feist*: the issue of whether a compilation that admittedly contains expressive material is disqualified from protection because it performs a useful function.

3. The 1-2-3 Menu Command Hierarchy as a Computer Program

The preferable approach to the question of classification of authorship in the 1-2-3 menu command hierarchy is to analyze the hierarchy as a computer program. This result may be accomplished in one of two ways.

First, under traditional principles, where an integral portion of a work has been copied, one analyzes the protectibility of the portion appropriated according to the nature of the work as a whole. The 1-2-3 menu command hierarchy is an integral part of the larger 1-2-3 computer program: it is not marketed separately from the program and has no existence separate from the program. The menu command hierarchy may, therefore, properly be analyzed under principles applicable to computer programs.

The second reason for analyzing the 1-2-3 menu command hierarchy as a computer program is that in one important respect it functions as a computer program, that is, as statements or instructions that are a subset of the larger 1-2-3 "set" and which themselves bring about a "certain result."[70]

As displayed on the computer screen, the menu command hierarchy acts as a menu of command choices, permitting the user to

tion, Judge Keeton's opinions in the district court contain references to the hierarchy as a compilation.

> [L]ike Lotus's menu tree, the protected expression of a compilation may be viewed as a system for accessing information. The purpose of a compilation is to communicate facts. . . . The selection, arrangement, and manner of presentation in a compilation may provide the user with a *method* or systematic manner of accessing the (uncopyrightable) facts. Thus, copyright law protects only that part of a compilation that the reader actually *uses* for selection of facts that the reader wants to know. Nevertheless, the expressive aspects of a compilation remain copyrightable.
>
> Accordingly, I conclude that the fact that the copied menu structure and first letters of the menu commands may be used to specify executable operations does not bar a finding that these elements are copyrightable. For the reasons specified in detail in Sections II.A.1. and II.B.2.b of the Phase I opinion, the structure of the menu tree including its designated keys for invoking commands (*i.e.* what Borland copied into the phantom menus) may also be viewed, in a light favorable to Borland, as a type of selection and arrangement of the executable operations in Lotus 1-2-3.

Lotus Dev. Corp., 831 F. Supp. at 231.

> Even though the executable operations are not copyrightable, the menu tree is copyrightable because the (hierarchical) arrangement of the definition and identification of executable operations contains expression.

Lotus Dev. Corp., 831 F. Supp. at 211.

[70] *See infra* text accompanying note 139 (definition of "computer program").

select tasks which he or she wishes to perform. A selection of one task may lead the user to one of the remaining forty-nine menus or submenus, where additional selections are made. In this role, the menu command hierarchy acts as a computerized compilation.[71] It should also be noted, though, that all computer programs are a mixture of code and data since the code requires data to execute various functions. Thus, the argument that the menu command hierarchy is data and not code does not necessarily preclude it from being part of a computer program.[72]

Lotus's 1-2-3 menu command hierarchy, however, performs a second role, and it is this second role that is one of the most important parts of the *Lotus Development Corp.* litigation. Users of spreadsheet programs repeat a number of tasks, such as entering specific types of data. In light of this, Lotus's 1-2-3 menu commands, in conjunction with the function keys on the computer (the top line on the keyboard—"F1," "F2," and so on), permit users to create what are called "macros," an abbreviation for "macro-instruction." Instead of repeatedly typing the same sequence of commands, the user can designate a few specific steps that thereafter will cause the computer to execute automatically an entire sequence of commands, much like what happens with automatic speed-dialing on a telephone.[73] These user-created steps are called "macros."[74]

When Lotus's menu commands are being used in their role of executing a user's macros, the menu commands arguably are acting as a computer program; each menu command is an individual statement or instruction in the user's macro. Collectively, the commands are a set of statements or instructions that bring about "a certain result" in the computer: the execution of the user's macros. As one guide to Lotus 1-2-3 states, "[m]acros are really computer programs designed to perform . . . their tasks within the confines of the 1-2-3 spreadsheet."[75]

[71] In Digital Communications Assocs. v. Softklone Distrib. Corp., 659 F. Supp. 449, 462-63 (N.D. Ga. 1987), the court protected a computer "status screen" display (a display which contained an arrangement of command terms under various headings) as a compilation, even though the only registration was for the underlying computer program.

[72] This also is consistent with the Copyright Office's position in its original decision to register computer programs in 1964. *See infra* text accompanying notes 106-11.

[73] This execution also occurs because the 1-2-3 code "interprets" the macro command designations made by the user.

[74] The Lotus 1-2-3 menu commands can copy ranges of cells, erase numbers, and save files. They can be used in at least four ways: (1) *interactive* (permits the user to pause in an ongoing task to enter data, and then resume the task after the data is entered); (2) *flow control* (directs the macro to execute a specific set of statements or instructions or to another macro); (3) *file manipulation* (reads and writes to other files); and (4) *data manipulation* (controls how data is entered and edited).

[75] JOHN WEINGARTEN, MORE 1-2-3 FOR DOS FOR DUMMIES 185 (1994). Macros are not

It is the macro function that Borland copied—for business reasons—and which led the First Circuit (for nonlegal reasons) to find the work unprotectible; when users have created their own macros and want to switch to a different company's spreadsheet program, they cannot use their 1-2-3 macros with the competing spreadsheet program *unless* Lotus's menu command hierarchy has been copied by the competing spreadsheet. Why? Because it is Lotus's menu commands that execute the macro, acting as the "set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[76] In order for the user's macros to be executed by a competing company's spreadsheet program (like Borland's Quattro Pro), the competing company's spreadsheet program must be capable of interpreting Lotus's menu command hierarchy—the set of statements or instructions—and functioning as Lotus 1-2-3 would under like circumstances.[77] Given these facts, viewing the menu command hierarchy as "statements or instructions" is an appropriate approach.[78] Under this approach, principles governing computer programs generally should apply.

III. The Structure of the Copyright Act: The Relationship between Section 102(a) and the Section 101 Subject Matter Definitions

The structure of the 1976 Copyright Act is transparent, the result of twenty-one years of deliberation and refinement. Broad exclusive rights are granted in section 106, subject only to the limi-

entirely composed of menu commands. They may include "commands other than menu commands, and keystrokes having nothing to do with commands at all." Brief for Petitioner, *Lotus Dev. Corp.*, *supra* note 27, at 10 n.17.

One may argue that because it is the user who designates the macro, the user is the "author" or at least a "co-author" of the macros. Users can, however, only create macros specified by Lotus and executable by the 1-2-3 code. *See* Stern Elec., Inc. v. Kaufman, 669 F.2d 852, 856 (2d Cir. 1982) (Newman, J.) (rejecting argument that player of electronic audiovisual game was a co-author of the game).

[76] 17 U.S.C. § 101 (definition of "computer program"). Arguably, it is the macro that executes the menu commands.

[77]
> To interpret a macro, the [Borland] program must use the Lotus 1-2-3 menu structure. 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 the [command designated by the user] refers to a path through the menu tree to the specific executable operation that changes a cell . . . .

Lotus Dev. Corp., 831 F. Supp. at 230.

[78] *Cf.* Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994). The Fifth Circuit analyzed a compilation of input formats contained in a computer program's manual as a computer program, writing that "[i]t makes no difference to the formats' copyrightability whether we analyze them as springing from a computer program or from a user manual." *Id.* at 1342 n.9.

tations contained in sections 107 through 120. Protected subject matter is enumerated in section 102(a) subject only to the limitations in section 102(b) and section 103.

Section 102(a) grants protection to "original works of authorship," a deliberately undefined term.[79] In *Feist*,[80] the Supreme Court construed the term as meaning "only that the work was independently created by the author . . . and that it possesses at least some minimal degree of creativity."[81] While *Feist* teaches us what the requisite general level of creativity is under section 102(a), it does not inform us how that level may be reached for a particular class of work. The structure of the Copyright Act supplies us with the answer; the subject matter definitions in section 101 work in conjunction with section 102(a) by describing the nature of authorship and, therefore, how originality may be satisfied for the defined subject matter.[82]

Section 102(a) does not, though, list computer programs as a separate species of protected work. Instead, as discussed below in part IV, Congress regarded computer programs as "literary works," continuing a practice the Copyright Office developed under the 1909 Act. Given the absence of a separate enumeration of "computer program" in section 102(a), and the contemporaneous 1980 amendment of section 117[83] as a limitation on the scope of protection of computer programs, it may be argued that one is not required to examine the section 101 definition of "computer program" in determining their originality; that the only relevance of the definition is in connection with section 117.

This argument, while certainly plausible, overlooks the structure of the Copyright Act and the purpose of section 117, which is a limitation on the scope of protection for admittedly protected computer programs. Section 117 does not tell us when a computer program is an original work of authorship. The answer to that question, I submit, must be sought in the statute, in the complementary interaction between section 102(a) and the section 101 definition of "computer program."

This interaction also is seen in the case of "compilations," which are generally literary works, and which are also not sepa-

[79] "The phrase 'original works of authorship,' which is purposely left undefined, is intended to incorporate without change the standard of originality established by the courts under the present copyright statute." H.R. REP. No. 1476, *supra* note 10, at 51, *reprinted in* 1976 U.S.C.C.A.N. at 5664; S. REP. No. 473, *supra* note 10, at 50.

[80] 499 U.S. 340.

[81] *Id.* at 345.

[82] *Cf. supra* note 41.

[83] 17 U.S.C. § 117.

rately enumerated in section 102(a). Like computer programs, compilations are defined in section 101 and also have a separate section containing a limitation on their scope of protection, section 103.[84] In *Feist*, the Supreme Court followed the analysis set forth here. First, the Court addressed the global question of originality for all works under section 102(a).[85] Then, the Court applied its general standard of originality to the subject matter before it, compilations, by examining the section 101 definition of "compilation" in order to determine how originality is to be satisfied for that class of work.

The same analysis should be applied to computer programs. The fact that the section 101 definition of "computer program" was added at the same time as section 117 does not require a different result. There is no reason to believe the courts would have been hampered in the slightest way in their interpretation of section 117 by the absence of such a definition. For example, section 115 is a limitation on the scope of protection for nondramatic musical works embodied in phonorecords.[86] Yet, Congress has not defined "nondramatic musical works." Moreover, when Congress uses a term that has relevance only to a particular section, it consistently includes definitions in the section itself, rather than placing the definition in section 101. For example, section 111(f) contains definitions that apply to that section's cable compulsory license.[87] Section 118, a compulsory license for public broadcasting, contains a definition in section 118(g).[88] Section 119, a compulsory license for satellite transmissions, contains its own definitions in subsection (d).[89] Thus, if Congress had intended the definition of "computer program" to be relevant only to section 117, it would have followed this consistent practice and placed the definition in section 117. By instead placing the definition in section 101, Congress indicated its intent that like "compilations" and "audiovisual works," the definition of "computer program" would be applicable throughout the Act, including to answer the question of originality.

[84] 17 U.S.C. § 103(a). One may argue that § 103 is also an enumeration of protected subject matter, but this argument is refuted by the language of § 103(a), which begins "[t]he subject matter of copyright as specified by section 102," thereby indicating that § 102 is the sole source of subject matter protection. 17 U.S.C. § 103(a).

[85] The Court also decided the question of originality on constitutional grounds, equating "originality" with "author," the latter a constitutional term. *Feist*, 499 U.S. at 345-46.

[86] 17 U.S.C. § 115.

[87] 17 U.S.C. § 111(f).

[88] 17 U.S.C. § 118(g).

[89] 17 U.S.C. § 119(d).

IV. Computer Programs as a Type of Literary Work

A. *Overview*

The lower courts in computer program cases have ignored the above-discussed structure of the Act—the interaction between section 102(a) and the section 101 definition—and instead have analyzed computer programs under the larger genus of "literary work." In *Lotus Development Corp.*, petitioner Lotus similarly is arguing that the Supreme Court should analyze computer programs "as literary works, subject to the same principles applicable to other, more familiar works of authorship."[90]

Ironically, even though "high protection" courts such as the Third Circuit[91] and "low protection" courts such as the Second Circuit[92] differ dramatically in their approaches to copyright in computer programs, both ground their decisionmaking on the following false syllogism:

1. Plays are "literary works;"
2. Copyright in plays may extend beyond the literal words to detailed plots and characters;[93]
3. Computer programs are classified as "literary works;"
4. Therefore, copyright in computer programs may extend beyond literal code to their detailed plot—structure, sequence, and organization—and must be analyzed according to the same principles as are plays.

The appeal to treat computer programs "under the same principles applicable to other literary works"[94] ignores two critical considerations. First, there are *no* principles generally applicable to the category of "literary works." The category broadly encompasses computer programs, compilations of data, newspapers, newsletters, magazines, legal treatises, histories, biographies, "how-to" books, almanacs, encyclopedias, insurance policies, tests, travel guides, cookbooks, language books, instruction manuals for consumer appliances, novels, plays, short stories, children's books, poetry, a vast array of business documents, liner notes on records, law review articles, and letters,[95] to name only a few. Second, as reviewed imme-

[90] Brief for Petitioner, *Lotus Dev. Corp.*, *supra* note 27, at 16.

[91] *See infra* text accompanying notes 216-22.

[92] *See infra* text accompanying notes 223-43.

[93] Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931).

[94] *See supra* note 44 (question presented for certiorari).

[95] Even the category of "letters" is diverse, including highly stylized literary correspondence, love notes, letters to the editor of a newspaper, complaints to government officials, and written exchanges between Supreme Court Justices about pending cases. In the con-

diately below, the classification of computer programs as "literary works" under the 1976 Copyright Act was the result—or accident—of Copyright Office administrative practices under the 1909 Act, rather than a belief that computer programs share inherent characteristics with plays or novels.

Finally, the presence of a special definition of "computer program" in section 101 indicates Congress intended protection for computer programs to be analyzed according to that definition, just as the definition of "compilation" in section 101 led the Supreme Court in *Feist* to analyze compilations according to that definition. Most compilations are also classified as "literary works," yet, it has not been suggested that they too must be analyzed like plays according to the above syllogism, nor did the Supreme Court in *Feist* resort to "the same principles applicable to other literary works" in analyzing compilations.[96] For purposes of statutory analysis, computer programs are no different than compilations in this respect.

Computer programs, though, because they must bring about a "certain result" in a machine, are different from plays. In a play, there are no restraints on the ordering or sequence of constituent elements, as is the case with computer programs (due to hardware constraints or programming language syntax). Published plays, while intended to be read, need not be logical nor particularly intelligible. By contrast, the code of computer programs is a form of "communication" only intelligible within the computer, and must conform to very precise criteria.

Nevertheless, because of the considerable ink that has been spilled on the issue of protecting computer programs "just like other literary works," and because the petitioner in *Lotus Development Corp.* asserted, erroneously, that Congress intended computer programs to be analyzed according to the "same principles as other literary works,"[97] it is well to review the history of protection of lit-

text of fair use claims, Judges Leval and Oakes of the Second Circuit cautioned against an unthinking tendency to lump diverse works together merely because they are unpublished. *See* Pierre N. Leval, *Toward a Fair Use Standard*, 103 HARV. L. REV. 1105, 1116-17 (1990); *Fair Use and Unpublished Works: Joint Hearing on S. 2370 and H.R. 4263 Before the Subcomm. on Patents, Copyrights and Trademarks and the Subcomm. on Courts, Intellectual Property, and the Administration of Justice of the House Comm. on the Judiciary*, 101st Cong., 2d Sess. 85 (1990) (supplementary statement of Hon. James L. Oakes). The same caution applies to the substantially larger class of all literary works. *See* Webb v. Powers, 29 F. Cas. 511, 516 (C.C.D. Mass. 1847) (No. 17,323) ("[T]here is much discrimination to be used in inquiries of this character, between different kinds of books, some of which, from their nature, cannot be expected to be entirely new.").

[96] *See supra* note 44 (question presented for certiorari).

[97] *See supra* note 44 (question presented for certiorari).

erary works.

B. *1790 to 1977: Protection of "Books"*

The first U.S. Copyright Act[98] extended protection to any "map, chart, book or books."[99] The term "book" was not defined, but was construed quite broadly: it extended not just to bound narrative works, but also to works that were compilations of information. For example, the Librarian of Congress's 1897 annual report to Congress noted: "[U]nder the head of books entered for copyright are classified hotel registers, form books, circulars, syndicate articles, and so on—the law providing no other term."[100] Members of Congress agreed,[101] as did the courts.[102] Indeed, although musical compositions did not receive express statutory protection until 1831, they too were registered under the 1790 Copyright Act as books.[103]

C. *Protection of Computer Programs as "Books" under the 1909 Copyright Act*

This practice of considering everything that did not fit into another category as a "book" continued under the Copyright Act of

[98] Act of May 31, 1790, ch. 15, 1 Stat. 124 (1790).

[99] Act of May 31, 1790, ch. 15, §§ 1-5, 1 Stat. at 124-25.

[100] REPORT OF THE LIBRARIAN OF CONGRESS, S. DOC. NO. 13, 55th Cong., 2d Sess. 5 (1897).

[101] *See* 42 CONG. REC. 981 (1896) (remarks of Mr. Cowan) ("[I]n the construction of this act the word 'book' is to be construed to mean every volume and part of a volume, together with all maps, prints, or engraving belonging thereto.").

[102] *See* Brightley v. Littleton, 37 F. 103, 104 (C.C.E.D. Pa. 1888) ("The statute . . . has been so liberally construed as to make it embrace within the term 'book,' every character of publication; whether a volume, pamphlet, newspaper article, calendar, or catalogue. In this construction our courts have simply followed those of England in their interpretation of similar language contained in the English statute."); Drury v. Ewing, 7 F. Cas. 1113, 1115 (C.C.S.D. Ohio 1862) (No. 4,095) (chart entitled "[t]he ladies' chart for cutting dresses and basques for ladies, and coats, jackets, etc., for boys," consisting of a single large sheet representing a series of diagrams interspersed with printed instructions, characterized as a "book" within the meaning of the statute); *see also* Holmes v. Hurst, 174 U.S. 82, 89 (1899) ("[T]he word 'book' as used in the statute is not to be understood in its technical sense of a bound volume, but any species of publication which the author selects to embody his literary product."). *But see* Amberg File & Index Co. v. Shea Smith & Co., 78 F. 479 (C.C.N.D. Ill. 1896), *aff'd*, 82 F. 314 (7th Cir. 1897) (individual who had created a system of 30 indexes, each index covering a letter or portion of a letter of the alphabet, was held not to be "author" or to have created a "book").

[103] *See* Clayton v. Stone, 5 F. Cas. 999, 1000 (C.C.S.D.N.Y. 1829) (No. 2,872) ("A book within the statute need not be a book in the common and ordinary acceptation of the word, viz., a volume made up of several sheets bound together; it may be printed only on one sheet, as the words of a song or the music accompanying it."). The first registration for a musical composition was made on January 6, 1794 for *The Kentucky Volunteer: A New Song*, written by "a lady of Philadelphia." The composer is believed to have been Raynor Taylor. *See* FEDERAL COPYRIGHT RECORDS 1790-1800 15 (James Gilreath ed., 1987).

1909,[104] and explains why, in 1964, when the Copyright Office decided to register computer programs,[105] it classified them as "books."

The Copyright Office's decision was the result of an application submitted on November 30, 1961 by North American Aviation, Inc., for a computer program (SCOPAC-PROG.63) embodied in magnetic tape. Subsequently, on April 20, 1964, a very brief computer program created by John Banzhaf, a student at Columbia University Law School, was published in the Columbia Law Review. Registration for this program was made in May 1964. Registration for the North American Aviation program was made in June 1964.[106]

The Copyright Office had doubts whether computer programs could be considered the "writing" of an "author" under the Constitution, and whether a machine-readable version of a program qualified as a "copy" under the 1909 Act.[107] Nevertheless, based on a "rule of doubt"—a policy of resolving doubtful cases in favor of the applicant—the Office announced it would permit registration of a computer program as a "book" if:

> (1) The elements of assembling, selecting, arranging, editing, and literary expression that went into the compilation of the computer program [were] sufficient to constitute original authorship.

[104] Act of Mar. 4, 1909, ch. 320, 35 Stat. 1075 (1909).

[105] *See* George Cary, *Copyright Registration and Computer Programs*, 11 BULL. COPYRIGHT SOC'Y 362 (1964); *Copyright Office Circular No. 61* (1964), *reprinted in* 11 BULL. COPYRIGHT SOC'Y 361 (1964).

The first "electronic computer"—a digital computational device, using binary numbers (base-two) rather than the traditional base-10 numbers, and separating data processing functions from memory functions—was invented in 1939 by Iowa State University Professor John V. Atanasoff, who thought of the invention while drinking a bourbon and water in a roadhouse. Nevertheless, for many years, J. Presper Eckert and John W. Mauchly, the inventors of the ENIAC (Electronic Numerical Integrator and Computer, patented in the 1940s after having allegedly been conceived over ice cream and coffee in a Philadelphia restaurant), were considered the fathers of the first electronic computer. Mauchly, however, previously had been extensively briefed by Professor Atanasoff on his work. In Sperry Rand v. Honeywell, 182 U.S.P.Q. (BNA) 673 (D. Minn. 1973), the court set history aright, invalidating the Eckert-Mauchly patent on the ground that it was based on Atanasoff's work. Atanasoff had made a prototype of his invention, but after unsuccessfully attempting to get the Remington Rand Corporation to build it, went on to other interests. *See generally* ALICE R. BURNS & ARTHUR W. BURNS, THE FIRST ELECTRONIC COMPUTER: THE ATANASOFF STORY (1988); CLARK R. MOLLENHOFF, ATANASOFF, FORGOTTEN FATHER OF THE COMPUTER (1988). Professor Atanasoff died on June 15, 1995 at the age of 91. *See* Claudia Levy, *John V. Atanasoff Dies at Age 91; Invented First Electronic Computer*, WASH. POST, June 19, 1995, at B4 (obituary). The first personal computer, the MITS Altair, was built by the MITS company of Albuquerque, New Mexico in 1975. Bill Gates, Chairman of Microsoft, was one of the computer programmers employed by MITS.

[106] *See* Cary, *supra* note 105, at 363.

[107] *Id.*

> (2) The program [was] published with the required copyright notice; that is, "copies" (i.e., reproductions of the program in a form perceptible to the human eye) bearing the notice [were] distributed or made available to the public.[108]
>
> (3) The copies deposited for registration consist of or include reproductions in a language intelligible to human beings. If the only publication was in a form that cannot be perceived visually or read, something more (e.g., a print-out of the entire program) would also have to be deposited.[109]

The Copyright Office defined a computer program as "either a set of operating instructions for a computer or a compilation of reference information to be drawn upon by the computer in solving problems."[110] Registration was believed appropriate since "substantial elements of gathering, choosing, rejecting, editing, and arranging material"[111] were involved in the creation of programs.

Two years later, in reporting a revision bill, the House Judiciary Committee adopted the Copyright Office's position, making clear (in language incorporated ten years later in the committee report that accompanied the 1976 Act[112]) that it regarded computer programs as a "new expressive form[ ]" of authorship that "could be regarded as an extension of copyrightable subject matter Congress had already intended to protect, and . . . thus considered copyrightable from the outset without the need of new legislation."[113]

D. *The 1976 Copyright Act: Protection of "Literary Works"*

In the 1976 Act, the category "books" became "literary works," and computer programs, along with scientific and technical works, charts, encyclopedias and other reference works, newspapers, magazines, pamphlets, letters, diaries, tests, instructional texts, business documents, compilations of information and data, law digests, histories, biographies, novels, and plays, were classified as "literary works."

The switch from "book" in the 1909 Copyright Act to "literary work" in the 1976 Copyright Act has a very prosaic explanation: the term "book" confuses the material object with the intangible sub-

[108] *See Supplementary Practice No. 35: Position of the Copyright Notice on Computer Programs of Domestic Origin First Published in Machine Readable Tape or Machine Punched Cards, in* COMPENDIUM I OF COPYRIGHT OFFICE PRACTICES S-125 (1968).

[109] *Copyright Office Circular No. 61, supra* note 105, at 361.

[110] *Id.*

[111] *Id.*

[112] *See infra* text accompanying note 119.

[113] H.R. REP. NO. 2237, 89th Cong., 2d Sess. 43 (1966).

ject matter embodied therein.[114] A play is a play regardless of whether it is printed in a hardbound edition (a "book"), in a magazine, on microfilm, or stored in the memory of a computer. To avoid this confusion between the intangible work of intellectual property and its tangible fixation, and perhaps to avoid the 1909 Act's repetitious statutory language regarding various forms of fixation,[115] the following definition of "literary works" was provided:

> works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.[116]

A Copyright Office report, issued in 1965 shortly after the definition's initial inclusion in revision bills, indicates that the defini-

[114] *See* House Comm. on the Judiciary, 88th Cong., 2d Sess., Copyright Law Revision Part 3: Preliminary Draft Law for Revised U.S. Copyright Law and Discussions and Comments on the Draft 54-55 (Comm. Print 1964) (comments of Abe Goldman, General Counsel, U.S. Copyright Office) [hereinafter Copyright Law Revision Part 3].

[115] *Id.* at 51, 54 (remarks of Mr. Goldman); *cf.* Act of Mar. 4, 1909, ch. 320, § 5(a)-(c), 35 Stat. at 1076.

[116] 17 U.S.C. § 101. A definition was first proposed in a 1963 preliminary draft bill prepared by the Copyright Office. This draft would have protected "[n]ondramatic literary works, including works expressed in words, numbers, or symbols representing them." Copyright Law Revision Part 3, *supra* note 114, at 54, § 1(a)(1). Although in passing one commentator opined that the phrase "number or symbols" would encompass storage in computer memory, *id.* at 52 (remarks of George Schiffer), the Copyright Office's General Counsel indicated that the purpose for inserting "words, numbers, or symbols" following nondramatic works was to create a broad, illustrative category without having to specify, as the 1909 Copyright Act had, "books, periodicals, and lectures." *Id.* at 51, 54.

The first revision bills, House Bill 11947, House Bill 12354, and Senate Bill 3008, introduced in 1964, would have changed the category of protected works simply to "literary works," enumerated in § 1(1). A separate definitions section, § 54, included a more extensive definition of "literary works" than the 1963 preliminary draft bill.

"Literary works" were defined 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, manuscripts, phonorecords, or film, in which they are embodied.

While this definition had the virtue of making a clear distinction between the intangible intellectual property (the "literary work") and its possible fixations (in "material objects" such as books), it was not appreciably more illuminating than the terse 1963 reference.

The 1964 definition is identical to the definition found in § 101 of the 1976 Copyright Act, except that the 1964 definition did not list "tapes, disks, or cards" as material objects in which the literary work could be embodied. *See* House Comm. on the Judiciary, 89th Cong., 1st Sess., Copyright Law Revision Part 5: 1964 Revision Bill With Discussions and Comments 30 (Comm. Print 1965). These words were added, without explanation, on Sept. 3, 1976, in the revision bill as reported by the House Judiciary Committee. *See* H.R. Rep. No. 1476, *supra* note 10, at 3. It appears that these additional terms were derived from a proposed definition of "computer program" submitted by the Information Industry Association ("IIA") at a May 7, 1975 hearing before the copyright subcommittee of the House Judiciary Committee. *See Copyright Law Revision: Hearing on H.R. 2223 Before the Subcomm. on Courts, Civil Liberties, and the Administration of Justice of the House Comm. on the Judiciary*, 94th Cong., 1st Sess. (1975) [hereinafter *Hearing on H.R. 2223*]. *See infra* text accompanying note 127.

tion is intended to be broad enough "to cover every possible form of verbal or numerical expression including, for example, computer programs fixed on punchcards, magnetic tape, or any other media."[117]

The relevant language from the 1966 House Judiciary Committee report, incorporated in the legislative reports accompanying the 1976 Act, is:

> The history of copyright law has been one of gradual expansion in the types of works accorded protection . . . . [S]cientific discoveries and technological developments have made possible new forms of creative expression that never existed before. In some of these cases the new expressive forms—electronic music, filmstrips, and computer programs, for example—could be regarded as an extension of copyrightable subject matter Congress had already intended to protect, and were thus considered copyrightable from the outset without the need of new legislation.[118]
>
> The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. It also 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.[119]

The phrase "qualitative limitations" is potentially misleading since "qualitative" may be taken to refer to a judgment regarding aesthetics or creativity, a judgment that is equally applicable to plays or novels. A more plausible explanation is that the definition does not intend any *subject matter* limitations within the field of "literary works": all original works fitting within the definition are covered.

E. *The National Commission on New Technological Uses of Copyrighted Works*

Beginning in 1967, efforts were made to establish a commission (a classic Washington solution) to study the issue of infringement of traditional copyrighted works by computers. Those efforts

[117] House Comm. on the Judiciary, 89th Cong., 1st Sess., Copyright Law Revision Part 6: Supplementary Report of the Register of Copyrights on the General Revision of the U.S. Copyright Law: 1965 Revision Bill 5 (Comm. Print 1965).

[118] H.R. Rep. No. 1476, *supra* note 10, at 51, *reprinted in* 1976 U.S.C.C.A.N. at 5664; S. Rep. No. 473, *supra* note 10, at 50.

[119] H.R. Rep. No. 1476, *supra* note 10, at 54, *reprinted in* 1976 U.S.C.C.A.N. at 5667-68; *see* S. Rep. No. 473, *supra* note 10, at 53.

were not successful until December 31, 1974,[120] when a law establishing the National Commission on New Technological Uses of Copyrighted Works ("CONTU") was signed by President Ford.[121] CONTU was charged with studying and making recommendations on the following issues:

> (1) the reproduction and use of copyrighted works of authorship—
>
> (A) in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information, and
>
> (B) by various forms of machine reproduction, not including reproduction by or at the request of instructors for use in face-to-face teaching activities; and
>
> (2) the creation of new works by the application or intervention of such automatic systems or machine reproduction.[122]

Conspicuously absent is any authority or direction for CONTU to study issues concerning whether copyright should be extended to computer programs. Instead, CONTU's focus was to be on infringement of traditional works *by* computers. The reason for the absence of a mandate to study issues concerning whether copyright should be extended to computer programs is obvious; Congress[123] had already concluded they were protected under the then-governing 1909 Copyright Act and that they would continue to be protected under the new act.

On July 31, 1978, following three years of study, CONTU submitted its report and recommendations to President Carter.[124] In the meantime, the 1976 Copyright Act had been passed and had become effective on January 1, 1978. In light of the fact that its mandate did not include the question of software protection, it is surprising that CONTU's report focused almost entirely on the questions of copyrightability and scope of protection for computer programs.

[120] *See* Final Report of the National Commission on New Technological Uses of Copyrighted Works 3-4 (1979) [hereinafter CONTU Final Report]. For a reference to computer programs at hearings before the Senate in 1967, see *Copyright Law Revision: Hearings on S. 597 Before the Subcomm. on Patents, Trademarks, and Copyrights of the Senate Comm. on the Judiciary*, 90th Cong., 1st Sess. 192-201 (1967) (testimony of Prof. Arthur Miller).

[121] Act of Dec. 31, 1974, 88 Stat. 1873 (1974).

[122] Act of Dec. 31, 1974, § 201(b), 88 Stat. at 1873.

[123] Although the House Judiciary Committee issued the 1966 report (quoted *supra* text accompanying note 119), all Senate reports during the revision process incorporated the same language verbatim. *See* S. Rep. No. 983, 93d Cong., 2d Sess. 104 (1974); S. Rep. No. 473, *supra* note 10, at 50. Of course, to be accurate, "Congress" cannot be said to have acted until both houses passed the legislation in 1976.

[124] *See* CONTU Final Report, *supra* note 120.

The majority of CONTU's commissioners recommended that the Copyright Act be amended "to make explicit that computer programs, to the extent that they embody an author's original creation, are the proper subject matter of copyright . . . ."[125] The commissioners also proposed that the following definition be added to section 101 of the Copyright Act:

> A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.[126]

No explanation of the definition was provided, and it was subsequently incorporated into the Copyright Act, also without explanation. The origin of the definition does not lie with CONTU, though, but with testimony submitted by the Information Industry Association ("IIA") before the House Subcommittee on Courts, Civil Liberties, and the Administration of Justice on May 15, 1975. The IIA submitted the following definition, in which, despite its length, one can perceive the basic outline of what was to become the section 101 definition of "computer program":

> A "computer program" is a literary work consisting of a *series of instructions o[r] statements* which are *in a* form acceptable to a *computer* and which are prepared *in order to achieve a certain result,* regardless of the nature of the material objects, such as documents, punched cards, magnetic tapes or discs, or computer storage elements, in which the works are embodied. A computer program may be a derivative work of a flow chart and either may be a derivative work of a literary work.[127]

A year later, at a May 7, 1976 meeting of CONTU, the IIA proposed the identical definition.[128] The essence of what became the section 101 definition[129] can be distilled from the IIA's proposal by dropping out redundant language:

> A "computer program" is . . . a series of instructions o[r] statements which are in a form acceptable to a computer and which are prepared in order to achieve a certain result.

Deletion of such redundant material is, in fact, the approach taken by the CONTU's own software subcommittee in its report[130]

[125] *Id.* at 1.

[126] *Id.* at 12.

[127] *Hearing on H.R. 2223, supra* note 116, at 333 (emphasis supplied).

[128] *See* Transcript CONTU Meeting Number 6, NTIS PB-254 765, at 122 (Dep't Commerce May 1976).

[129] *See infra* text accompanying note 139.

[130] *Report of the Software Subcommittee to the National Commission on New Technological Uses of*

to CONTU in the summer of 1977:

> *Proposed New Definition:*
>
> The Subcommittee believes that it would be appropriate to include among the definitions in § 101 the definition of a computer program. We suggest the following language:
>
> > "A 'computer program' is a fixation of a series of statements or instructions to be used in conjunction with a computer in order to bring about a certain result."[131]

The CONTU software subcommittee's language is strikingly similar to a stripped-down version of that presented by the IIA to Congress. Few comments were submitted on the software subcommittee's proposal.[132] One comment, by Computer & Business Equipment Manufacturers Association ("CBEMA"), should be mentioned because it proposed a substitute definition, the italicized elements of which appear in the final definition in section 101 of the Copyright Act:

> A "computer program" is a literary work that consists of a series of statements or instructions in a form and order intended to be *directly or indirectly* utilized in a computer to cause it to achieve a certain result through the performance of a series of steps of a process.[133]

CBEMA explained that "indirectly" was proposed because many programs are written in source code and need to be compiled or assembled into object code before they can be "directly" used by the computer.[134] There are no further references in the transcripts of CONTU meetings explaining how the final version of the definition of "computer program" came about.

F. *The 1980 Software Amendments*

In 1980, an amended version of CONTU's recommendations was included in House Bill 6934, a patent revision bill.[135] Brief testimony on the copyright parts of the bill was given before the House Subcommittee on Courts, Civil Liberties, and the Adminis-

Copyrighted Works, reprinted in Transcript of 16th CONTU Meeting, NTIS PB-273 594, at 183-208 (Dep't Commerce Sept. 1977).

[131] *Id.* at 198.

[132] *See id.* at 18-19 (rather incomprehensible comment by representative of Association of Computing Machinery); *id.* at 233 (Commissioner Hersey's proposed definition) ("A 'computer program' is a set of machine directions designed to control the operations of a computer in order to obtain a desired result.").

[133] *Id.* at 78 (emphasis supplied).

[134] *Id.* at 79.

[135] H.R. 6934, 96th Cong., 2d Sess. (1980).

tration of Justice on April 15[136] and May 8, 1980.[137] On December 12, 1980, President Carter signed the bill into law.[138] The legislation contained verbatim the definition of "computer program" proposed by CONTU: "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[139]

One point that should be noted about this definition is that, by limiting "computer program" to a set of statements or instructions to be used in a computer in order to bring about a "certain result," the definition excludes manuals, flow charts, and other representations that cannot be so used.[140]

The regime for computer program copyright protection that finally emerged in 1980 (and which governs today) is the product of sixteen years of activity in the Congress, the Copyright Office, and CONTU. That regime is: administrative classification as a "literary work" under 102(a), with a definition of the nature of authorship (and therefore originality) under section 101, but subject to limitations on protection in sections 117 and 102(b). We now turn to section 102(b), due to the importance it has been accorded in recent decisions, particularly *Lotus Development Corp. v. Borland International, Inc.*

V. The Idea-Expression Dichotomy and Section 102(b) of the 1976 Copyright Act

A. *The Function of Section 102(b)*

While section 102(a) extends protection to all "original works of authorship,"[141] the scope of that protection is subject to an important limitation in section 102(b):

> In no case does copyright protection for an original work of au-

[136] *Industrial Innovation and Patent and Copyright Law Amendments: Hearings on H.R. 6933 et al. Before the Subcomm. on Courts, Civil Liberties, and the Administration of Justice of the House Comm. on the Judiciary*, 96th Cong., 2d Sess. 42-46 (1980).

[137] *Id.* at 635-36, 683-86, 698-701.

[138] Act of Dec. 12, 1980, § 10(b), 94 Stat. 3015, 3028 (1980); *see* H.R. Rep. No. 1307, pt. 1, 96th Cong., 2d Sess. 23-24 (1980); H.R. Rep. No. 1307, pt. 2, 96th Cong., 2d Sess. 19 (1980).

[139] 17 U.S.C. § 101.

[140] Source code is encompassed within the definition because, through use of a compiler or assembler, it can be used in a computer. For purposes of registration, the source or object code versions of a work are treated as one work. *See* Compendium II of Copyright Office Practices § 321.03 (1984); Data General Corp. v. Grumman Sys. Support Corp., 803 F. Supp. 487 (D. Mass. 1992), *aff'd*, 36 F.3d 1147 (1st Cir. 1994); GCA Corp. v. Chance, 217 U.S.P.Q. (BNA) 718 (N.D. Cal. 1982); Hudson v. Good Rush Messenger Serv., Inc., 1987-1988 Copyright L. Dec. (CCH) ¶ 26,089 (S.D.N.Y. Apr. 16, 1987).

[141] 17 U.S.C. § 102(a).

> thorship 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.[142]

Section 102(b) assumes the existence of a protected work—"an original work of authorship"—but cautions that protection for such an original work of authorship does not extend to any unprotectible components contained therein. As the Supreme Court explained in *Feist*:

> The mere fact that a work is copyrighted does not mean that every element of the work may be protected. Originality remains the *sine qua non* of copyright; accordingly, copyright protection may extend only to those components of a work that are original to the author.[143]

Section 102(b) is but another indication of this basic principle; it enumerates illustrative components that are not subject to protection, and which, therefore, may be appropriated without infringing the work in which they are embodied.[144] Read one way, though, section 102(b) might be interpreted to preclude copyright in entire works, including a great deal of material ordinarily considered to be subject to protection, such as an original illustration of the idea of a myopic New Yorker's view of the world in the form of a fanciful map of Manhattan and the world.[145] The purpose of section 102(b), however, is not to deny protection entirely to such works, but instead to "restate . . . that the basic dichotomy between

[142] 17 U.S.C. § 102(b); *see* 37 C.F.R. § 202.1(b) (1994) ("Ideas, plans, methods, systems, or devices [are not copyrightable], as distinguished from the particular manner in which they are expressed or described in a writing . . . ."); *Copyright Office Circular* No. 31 ("Copyright protection is not available for: ideas or procedures for doing, making, or building things; scientific or technical methods or discoveries; business operations or procedures; mathematical principles; formulas or algorithms; or any concept, process, or method of operation.").

[143] *Feist*, 499 U.S. at 348.

[144] This is also the Supreme Court's interpretation of § 102(b). *See Feist*, 499 U.S. at 356 (section 102(b) "identifies specifically those elements of a work for which copyright is not available," and tracing history of the section to § 3 of the 1909 Copyright Act, which stated that copyright extended only to the "copyrightable parts" of the work).

The strangest interpretation of § 102(b) may be that of the Nimmers, who argue that the distinction between idea and expression drawn by the section "constitutes not so much a limitation on the copyrightability of works, as it is a measure of the degree of similarity which must exist as between a copyrightable work and an unauthorized copy, in order to constitute the latter an infringement." MELVILLE B. NIMMER & DAVID NIMMER, 1 NIMMER ON COPYRIGHT § 2.03[D], at 2-34 (1995) [hereinafter NIMMER ON COPYRIGHT]. This interpretation misses the whole point of § 102(b): *any* similarity between two works resulting from the copying of ideas, methods of operation, etc. is not actionable because any such material is, *ab initio*, unprotected.

[145] *See* Steinberg v. Columbia Pictures Indus., 663 F. Supp. 706 (S.D.N.Y. 1987).

expression and idea remains unchanged."[146] Register of Copyrights Abraham Kaminstein explained this dichotomy in a 1961 report:

> Copyright does not preclude others from using the ideas or information revealed by the author's work. It pertains to the literary, musical, graphic, or artistic form in which the author expressed intellectual concepts. It enables him to prevent others from reproducing his individual expression without his consent. But anyone is free to create his own expression of the same concepts, or to make practical use of them, as long as he does not copy the author's form of expression.[147]

Thus, while anyone can illustrate his own idea of a New Yorker's (or Washingtonian's) myopic view of the world, he is not free to copy Saul Steinberg's illustration of that idea.

Although section 102(b) applies to all forms of subject matter, the legislative history reveals a strong connection to computer programs. Section 102(b) was first inserted in the 1969 Senate omnibus revision bill.[148] The first committee report after this insertion notes a

> concern . . . [that] copyright in computer programs should ex-

[146] H.R. Rep. No. 1476, *supra* note 10, at 57, *reprinted in* 1976 U.S.C.C.A.N. at 5670-71; S. Rep. No. 473, *supra* note 10, at 54.

[147] House Comm. on the Judiciary, 87th Cong., 1st Sess., Copyright Law Revision: Report of the Register of Copyrights on the General Revision of the U.S. Copyright Law 3 (Comm. Print 1961); *accord* H.R. Rep. No. 1476, *supra* note 10, at 56, *reprinted in* 1976 U.S.C.C.A.N. at 5669-70; S. Rep. No. 473, *supra* note 10, at 54; *Feist*, 499 U.S. at 356 (citing legislative reports). *See* Holmes v. Hurst, 174 U.S. 82, 86 (1899) ("The right thus secured by the copyright act is not a right to the use of certain words, because they are the common property of the human race, and are as little susceptible of private appropriation as air or sunlight; nor is it the right to ideas alone, since in the absence of means of communicating them they are of value to no one but the author."); Mazer v. Stein, 347 U.S. 201, 217 (1954).

The distinction between ideas and expression also is recognized internationally. *See* Agreement on Trade-Related Aspects of Intellectual Property Rights art. 9(2), *reprinted in* Message from the President of the United States Transmitting the Uruguay Round of Trade Agreements, Texts of Agreements Implementing Bill, Statement of Administrative Action and Required Supporting Statements, H.R. Doc. No. 316, 103d Cong., 2d Sess. 2 (1994) ("Copyright protection shall extend to expressions and not to ideas, procedures, methods of operation or mathematical concepts as such.").

> A fundamental point is that ideas, as such, are not protected by copyright. It is the patent rather than the copyright laws to which one must look for this protection. Subject, therefore, to patent protection, a person who has made his idea public has no means of stopping others [from] using it. But once that idea has been elaborated and expressed, copyright protection exists for the words, notes, drawings, etc., in which it is clothed. In other words, it is the form of expression which is capable of protection and not the idea itself.

World Intellectual Property Organization, Guide to the Berne Convention for the Protection of Literary and Artistic Works 12 (1978) (comment on art. 2(1) of the Berne Convention, defining "literary and artistic works").

[148] S. 543, 91st Cong., 1st Sess. (1969).

> tend protection to the methodology or processes adopted by the programmer, rather than merely to the "writing" expressing his ideas. Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied are not within the scope of the copyright law.[149]

The addition of section 102(b), though, had much less to do with the scope of copyright in computer programs than with infringement of traditional copyrighted works by virtue of their reproduction on computers. Users of copyrighted materials and authors and publishers, who had been fighting over this issue, reached a compromise in 1969. User groups previously had proposed a three-year moratorium on liability for copyright infringement by reproduction in computers. Authors and publishers vigorously opposed a moratorium, threatening to prevent passage of the entire revision bill if such a proposal was seriously considered. The compromise was enacted in a new section 117 that preserved the status quo on infringement issues pending a report by CONTU. Section 102(b) was inserted in Senate Bill 543[150] along with section 117.

The only statutory change in section 102(b) came in September 1976, when the House Judiciary committee deleted the word "plan" out of concern that architectural plans would be inadvertently excluded from protection.[151] The legislative report language

[149] S. Rep. No. 983, *supra* note 123, at 107. Technically, this report accompanied S. 1361, 93d Cong., 2d Sess. (1974), since the 1969 bill was not reported out by the Senate Judiciary Committee. Although it is common to construe the passage quoted in the text as referring to the nonprotectibility of algorithms, none of the legislative reports during the revision process uses the term. A 1975 draft Copyright Office report does, however, attribute the origin of § 102(b) in part to a desire not to protect algorithms used by programmers. *See* U.S. Copyright Office, Second Supplementary Report of the Register of Copyrights on the General Revision of the U.S. Copyright Law: 1975 Revision Bill, ch. I at 2 (draft Oct.-Dec. 1975) [hereinafter Draft Second Supplementary Report].

The European Union's Software Directive also takes into account the idea-expression dichotomy in the context of computer programs. *See* Council Directive 91/250 of 14 May 1991 on the Legal Protection of Computer Programs art. 1(2), 1991 O.J. (L 122) 42, 44 ("Ideas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright under this Directive."); *see also* Chosakuken-hō [Copyright Law], Law No. 48 of 1970, art. 10(3) (Japan), *amended by* Law No. 62 of 1985, *translated in* United Nations Educational, Scientific and Cultural Organization, Copyright Laws and Treaties of the World, at Japan: Item 1, at 6 (Supp. 1987-88) ("The protection granted by [the copyright] Law . . . shall not extend to any programming language, rule or algorithm used for making such works.").

[150] S. 543, *supra* note 148.

[151] *See* Draft Second Supplementary Report, *supra* note 149, ch. I at 6-7; *see also* S. Rep. No. 473, *supra* note 10, at 54 ("The term 'plan' in this context [§ 102(b)] refers to a mental formulation for achieving something, as distinguished from a graphic representation diagramming the mental concept.").

quoted above was incorporated in the legislative reports accompanying the 1976 Copyright Act without further explanation.[152]

As with many purely political compromises, the inclusion of section 102(b) was unnecessary. In the case of computer programs, the requirement of originality would have limited protection to the programmer's expression even in the absence of section 102(b); mathematical algorithms are no more copyrightable than the scientific principles courts easily had excluded from protection under the 1909 Act.[153]

By using undefined terms such as "process" and "method of operation," though, a potential conflict arose between section 102(b) and other sections of the Copyright Act when in 1980 the definition of "computer program" was added. The potential conflict is caused by the definition's requirement that the set of statements or instructions be used in a computer to bring about a "certain result."[154] If one considers the set of statements or instructions to be a "process" or "method of operation" because they bring about a certain result, then protection for all computer programs is eliminated.[155] Since Congress clearly indicated that computer programs are to be protected only if they *do* bring about "a certain result," the relationship between section 102(b) and the section 101 definition of "computer program" must be resolved in a way that gives meaning to both sections.

B. *Reconciling Section 102(b) and the Section 101 Definition of "Computer Program"*

The way to reconcile section 102(b)'s bar of protection to ideas, processes, and methods of operation and section 101's grant of protection to computer programs that bring about a certain result is two-fold and grounded in the language and structure of both sections. First, since originality in computer programs is to be construed according to the statutory definition as a set of statements or instructions,[156] the set itself cannot be regarded as a process or method of operation since this is the very manner in which Congress has instructed us that originality is to be satisfied. As the Second Circuit recently wrote in a copyright case: "A court should not read one part of a statute so as to deprive another part of mean-

[152] *See* H.R. Rep. No. 1476, *supra* note 10, at 57, *reprinted in* 1976 U.S.C.C.A.N at 5670-71.

[153] *See* Ricker v. General Elec. Co., 162 F.2d 141, 144-46 (2d Cir. 1947).

[154] *See supra* text accompanying note 139 (full definition).

[155] *See* Ginsburg, *supra* note 26, at 2570.

[156] *See infra* text accompanying notes 160-78.

ing."[157] Second, the language of section 102(b) directs the courts to exclude only the enumerated unprotectible *components* of the original work. In the case of computer programs, therefore, section 102(b) denies protection only to those components of the program's set of statements or instructions that constitute ideas, processes, or methods of operation.[158] Section 102(b), therefore, should not be used to deny protection to an entire set of statements or instructions which, as a set, contain more than a minimal amount of expressive material.[159]

VI. The Proper Approach to Determining Copyrightability of Computer Programs

A. *Step One: "Original Works of Authorship"*

Once section 102(b) is properly confined to excluding from protection only nonoriginal parts of a computer program, the question remains how originality is to be determined. The copyrightability of computer programs must, of course, be determined by the statute Congress has given us. In that statute, Congress has told us first that copyright extends to "original works of authorship."[160] This term was deliberately left undefined, with Congress stating a desire that the courts continue to apply the standard they already had developed.[161] In *Feist Publications, Inc. v. Rural Telephone Service Co.*, the Supreme Court authoritatively delineated that standard, holding that "[o]riginal, as the term is used in copyright, means only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity. . . . [a] creative spark . . . ."[162] The creative spark need not create a shock, but it must at least be perceptible to the touch.

[157] Carter v. Helmsley Spear, Inc., 71 F.3d 77, 85 (2d Cir. 1995) (citing United States Nat'l Bank of Or. v. Independent Ins. Agents of Am., Inc., 113 S. Ct. 2173, 2182 (1993)).

[158] This approach is consistent with the Supreme Court's interpretation of the origins of § 102(b). *See Feist,* 490 U.S. at 355-56; *see also supra* note 144.

[159] *See* M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421, 434-35 (4th Cir. 1986) (rejecting district court's ruling that plaintiff's computer video game was a "system" and finding that § 102(b) was intended merely to restate the idea-expression dichotomy).

Recognition that § 102(b) cannot be used to deny protection to an entire set of statements or instructions containing expressive elements eliminates the First Circuit's holding that the entirety of Lotus 1-2-3 is a method of operation. It also should be noted that the First Circuit's definition of method of operation, "the means by which a person operates something," *Lotus Dev. Corp.*, 49 F.3d at 815, is tautological since "means" is a synonym for "method" and "operates" is the verb form of "operation."

[160] 17 U.S.C. § 102(a).

[161] *See* H.R. Rep. No. 1476, *supra* note 10, at 51, *reprinted in* 1976 U.S.C.C.A.N. at 5664; S. Rep. No. 473, *supra* note 10, at 50.

[162] *Feist,* 499 U.S. at 345.

The originality standard is an objective one that does not permit the courts to use their own judgment of the aesthetic merit of the work, nor to cut back on or deny protection because the work serves a commercial purpose.[163] In place of such subjective evaluations, the only issue is whether the requirement of originality as construed by the *Feist* Court[164] has been met. In part, the disarray in the lower courts over computer programs is the result of some judges' antagonism to Congress's decision to protect computer programs under the copyright law. These judges view computer programs' functional and commercial attributes as worthy of protection, if at all, under the more rigorous requirements of the patent laws. Availability of protection under patent law is not, however, a bar to protection under the Copyright Act, as Congress made clear in that act.[165]

B. *Step Two: The Statutory Subject Matter Definitions*

While originality is an objective concept, it is also an abstract one. For although it informs us what the requisite general level of creativity is, it does not give us practical guidance about how that level can be reached in particular cases. The answer to how originality can be satisfied in particular cases is found instead in the definitions of subject matter Congress provided in section 101 of the Act.[166] As part II.C of the *Feist* opinion[167] and *Atari Games Corp. v. Oman*[168] recognized, these definitions express Congress's judgment about how, for the defined subject matter, the constitutional standard of originality can be satisfied.

A brief review of the definition of "compilation" as construed in *Feist* and the definition of "audiovisual work" as construed in *Atari Games Corp.* will demonstrate the methodology.

A "compilation" is defined as

> [A] work formed by the collection and assembling of preexisting materials or of data *that* are selected, coordinated, or arranged *in such a way that* the resulting work as a whole

[163] *See supra* text accompanying notes 4-8.

[164] *See supra* text accompanying note 162.

[165] *See* 17 U.S.C. § 301(d) ("Nothing in this title annuls or limits any rights or remedies under any other Federal statute."); *see also* U.S. PATENT AND TRADEMARK OFFICE & U.S. COPYRIGHT OFFICE, PATENT-COPYRIGHT LAWS OVERLAP STUDY x (1991) (prepared for the House Subcommittee on Intellectual Property and the Administration of Justice, and concluding that "there appears to be minimal overlap between the subject matter of copyrights and utility patents because the statutes make clear that the areas of protected matter in each case are markedly different").

[166] *But see supra* note 41.

[167] 499 U.S. at 354-61.

[168] 888 F.2d at 881, 979 F.2d 242. *See supra* note 21 for an explanation of case history.

constitutes an original work of authorship.[169]

The *Feist* Court construed the definition's tripartite structure (as indicated by the emphases) as indicating how originality could be satisfied. All three elements must be present: (1) there must be a collection of data, (2) from which the compiler selects, coordinates, or arranges certain data (3) in an original way. As the Court recognized,

> [t]he key to the statutory definition is the second requirement. It instructs courts that . . . they should focus on the manner in which the collected facts have been selected, coordinated, and arranged. . . . [T]he statute dictates that the principal focus should be on whether the selection, coordination, and arrangement are sufficiently original to merit protection.[170]

"Audiovisual works" are defined as

> works that consist of a series of related images which are intrinsically intended to be shown by the use of machines or devices . . . .[171]

Atari Games Corp. v. Oman involved the simple audiovisual electronic game "Breakout," in which no individual frame was protectible. The Copyright Office refused registration, ignoring thc work's original sequence of images. In reversing a district judge who had affirmed the Copyright Office's decision, Judge (now Justice) Ruth Bader Ginsburg instructed the Copyright Office to focus on "the audiovisual work as a whole, *i.e.*, the total sequence of the images displayed as the game is played."[172] When the Office again refused registration, Judge Ginsburg criticized the Office for analyzing only the individual screens "rather than the flow of the game as a whole."[173] She also cited the Supreme Court's intervening *Feist* opinion and found that *Feist*'s focus on the definition of "compilation" provided an analogous methodology for analyzing originality in audiovisual works. Both compilations and audiovisual works, she wrote, "involve a choice and ordering of elements that, in themselves, may not qualify for protection; the author's *selection* and *arrangement*, however, may 'entail [the] minimal degree of creativity' needed to bring the work within the protection of the copy-

[169] *Feist*, 499 U.S. at 356 (alteration in original) (quoting 17 U.S.C. § 101).

[170] *Id.* at 358.

[171] 17 U.S.C. § 101 (definition of "audiovisual work").

[172] *Atari Games Corp.*, 888 F.2d at 883.

[173] *Atari Games Corp.*, 979 F.2d at 245 (reviewing Copyright Office's decision on remand).

right laws."[174]

The definition of "computer program" bears a striking resemblance to that of "audiovisual work" as construed by Judge Ginsburg in *Atari Games Corp.*:

> Computer Program
>
> A "computer program" is a *set of statements or instructions to be used directly or indirectly in a computer* in order to bring about a certain result.[175]
>
> Audiovisual Works
>
> "Audiovisual works" are works that consist of a *series of related images which are intrinsically intended to be shown by the use of machines or devices* . . . .[176]

Both definitions direct attention to the whole rather than to the individual components.[177] Indeed, when one returns to the key second part of the definition of the "compilation"—the compiler's selection, coordination, or arrangement[178]—it is evident that the operative provisions of the definitions of "computer program," "compilation," and "audiovisual work" are quite similar. The term "set" in the definition of "computer program" equates with the term "series" in "audiovisual work," and with "select[ion and] . . . arrange[ment]" in "compilation."[179] All three definitions involve a collection of material where the building blocks may not be protectible; with compilations, the building blocks are data or other preexisting material; with audiovisual works, the building blocks are the individual frames; with computer programs, the building blocks are the individual statements or instructions. In all three definitions, though, originality may be found in the mere sequencing of the building blocks: the selection, coordination, or arrangement of data; the series of the images; and, the set of statements or instructions.

[174] *Id.* (quoting *Feist,* 499 U.S. at 348).

[175] 17 U.S.C. § 101 (definition of "computer program") (emphasis supplied). Although it is popularly believed that this definition originated with the National Commission on New Technological Uses of Copyrighted Works ("CONTU"), research for this article reveals that it originated with testimony submitted by the Information Industry Association before the House copyright subcommittee on May 15, 1975. *See Hearing on H.R. 2223, supra* note 116, at 333. It was not until a year later, at a May 7, 1976 meeting of the Commission, that the identical language was presented to the Commission. *See* Transcript CONTU Meeting Number 6, *supra* note 128, at 122; *see also supra* text accompanying notes 127-31.

[176] 17 U.S.C. § 101 (definition of "audiovisual work") (emphasis supplied).

[177] This does not mean original components cannot be protected, only that the absence of protectible components does not disqualify protection for the sequence.

[178] *See supra* text accompanying note 170.

[179] 17 U.S.C. § 101.

Thus, following *Feist* and *Atari Games Corp.*, so long as there is a "minimal degree of creativity" in the "choice and ordering" of the computer program's set of statements or instructions, the set as a whole is entitled to protection, even though any unoriginal component parts are not. This approach also fits well with section 102(b). The original aspects of a computer program are protected according to section 102(a) and the section 101 definition, while unoriginal components are excluded by section 102(b).

VII. False Paths

A. *The Futility of Bright-Line Tests*

Courts were faced with the argument that only the idea and not the expression of a work had been copied long before passage of the 1976 Copyright Act. After deciding copyright cases for almost fifty years, Judge Hand, in one of his final opinions, came to the conclusion that no general rule can be formulated for determining what is idea and what is expression, writing in *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*[180]: "Obviously, no principle can be stated as to when an imitator has gone beyond copying the 'idea,' and has borrowed its 'expression.' "[181] This difficulty should make courts particularly wary of announcing rules to be applied to classes of subject matter, including computer programs, since broad formulations are likely to be incapable of taking into account the wide varieties of creativity found in different types of authorship.

Judge Hand's statement in *Peter Pan Fabrics* stands in contrast to his more detailed review of the issue in *Nichols v. Universal Pictures Corp.*,[182] in which he contemplated various "patterns of generality" or "series of abstractions" that could be used to separate idea from expression in plays.

> Upon any work . . . a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the [work] is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the [author] could prevent the use of his "ideas," to which, apart from their expression, his property is never extended.[183]

[180] 274 F.2d 487 (2d Cir. 1960).

[181] *Id.* at 489. Chief Judge Newman recently noted the same difficulty in trade dress cases. *See* Jeffrey Milstein, Inc. v. Greger, Lawlor, Roth, Inc., 58 F.3d 27, 32 (2d Cir. 1995).

[182] 45 F.2d 119.

[183] *Id.* at 121.

Nichols assumes a unified, copyrightable work of fiction. The issue in *Nichols* was not copyrightability, but rather infringement where the defendant did not copy text, but, allegedly, characters and plot—integral parts of a unified whole. By contrast, in cases such as *Lotus Development Corp.*, the issue is copying of discrete textual elements.

Although some commentators have characterized Judge Hand's discussion in *Nichols* as an "abstractions test,"[184] and have successfully argued its adoption in computer program cases with unfortunate consequences,[185] as Judge Easterbrook wisely observed, the abstractions test "is not a 'test' at all. It is a clever way to pose the difficulties that require courts to avoid either extreme of the continuum of generality."[186] Apparently agreeing, Judge Keeton, citing *Peter Pan Fabrics*, declared in an early case involving Lotus's 1-2-3 spreadsheet program: "It seems the better part of wisdom, if not valor, not to press the search for a suitable bright-line test . . . where Learned Hand, even after decades of experience in judging, found none."[187]

Perhaps the better part of wisdom is to acknowledge that there is no idea-expression "test" either. The statement that given material is an (unprotectible) idea or (protectible) expression is merely a statement of the conclusion reached, rather than a methodology for reaching that conclusion. Whether given material is idea or expression must be made on an ad hoc basis taking into account the amount of originality involved, as well as any possible constraints on that originality. Some of those constraints may be technological. Because much emphasis has been placed in computer cases on technical constraints and the so-called "merger doctrine"—a specialized application of the idea-expression doctrine—it will be helpful to examine that doctrine.

B. *The Merger Doctrine*

The First Circuit's decision in *Lotus Development Corp.*[188]

[184] *See* 3 NIMMER ON COPYRIGHT, *supra* note 144, § 13.03[F], at 13-132.

[185] *See, e.g., Computer Assocs.*, 982 F.2d at 706-07. *See infra* text accompanying notes 223-43.

[186] Nash v. Columbia Broadcast Sys., 899 F.2d 1537, 1540 (7th Cir. 1990); *see* CCC Info. Servs., Inc. v. Maclean Hunter Mkt. Reports, 44 F.3d 61, 69 n.15 (2d Cir. 1994) (following *Nash*).

[187] *Paperback*, 740 F. Supp. at 60.

[188] 49 F.3d 807. *Lotus Development Corp.* is, though, in accord with earlier First Circuit precedent. In Morrissey v. Procter & Gamble Co., 379 F.2d 675 (1st Cir. 1967), the court of appeals wrote:

> When the uncopyrightable subject matter is very narrow, so that "the topic necessarily requires," if not only one form of expression, at best only a limited

notwithstanding, the idea-expression dichotomy should not preclude copyright in an entire work. This is also true of the related doctrine known as "merger."[189] The merger doctrine has been explained as follows by the Ninth Circuit:

> When the "idea" and its "expression" are . . . inseparable, copying the "expression" will not be barred, since protecting the "expression" in such circumstances would confer a monopoly of the "idea" upon the copyright owner free of the conditions and limitations imposed by the patent law.[190]

Subsequent courts have expanded the doctrine's reach, finding it applicable when there are also a few ways of expressing a particular idea.[191] The validity of merger as a doctrine separate from the idea-expression dichotomy is doubtful, however. If an

> number, to permit copyrighting would mean that a party or parties, by copyrighting a mere handful of forms, could exhaust all possibilities of future use of the substance. In such circumstances it does not seem accurate to say that any particular form of expression comes from the subject matter. However, it is necessary to say that the subject matter would be appropriated by permitting the copyrighting of its expression. We cannot recognize copyright as a game of chess in which the public can be checkmated.

Id. at 678-79 (citations omitted).

For other examples of courts finding entire works to constitute ideas, see Kern River Gas Transmission Co. v. Coastal Corp., 899 F.2d 1458, 1464 (5th Cir.), *cert. denied*, 498 U.S. 937 (1990); Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971); Production Contractors, Inc. v. WGN Continental Broadcasting Co., 622 F. Supp. 1500, 1503 (N.D. Ill. 1985) (idea of a Christmas parade "is a common one, relatively simple and contain[ing] no original creative authorship"); Arthur v. ABC, 633 F. Supp. 146, 148 (S.D.N.Y. 1985) (sketches of five interlocking rings, in which three of the rings were modified to represent the lower case letters a, b, and c held to be uncopyrightable as containing "no more than the bare idea or concept of superimposing . . . two logos," but sculptural work of the five rings found protectible); *see also* Op. Att'y Gen. of N.Y. No. 84-F9 (1984), 1985-1986 Copyright L. Dec. (CCH) ¶ 25,753, at 19,379 (ideas for how to redredge the Erie Canal or build a man-made island in New York Harbor not copyrightable); *cf.* G.D. Searle & Co. v. Philips-Miller & Assocs., 836 F. Supp. 520, 524 (N.D. Ill. 1993) (rejecting argument that advertising proposal was an idea). *But see* Mason v. Montgomery Data, Inc., 967 F.2d 135 (5th Cir. 1992) (limiting *Kern River*). In each of these cases, the courts would have been on sounder ground finding that the work lacked the requisite originality.

[189] *See Kern River*, 899 F.2d at 1463 ("The doctrine of 'merger' developed in an effort to deal with th[e] difficulty in locating the precise boundary between idea and expression.").

[190] *Kalpakian*, 446 F.2d at 742 (holding that a jewelled bee pin was a noncopyrightable idea). *Cf.* Herbert Rosenthal Jewelry Corp. v. Honora Jewelry Co., 509 F.2d 64 (2d Cir. 1974) (no infringement of jewelled turtle pins); Herbert Rosenthal Jewelry Corp. v. Grossbardt, 436 F.2d 315 (2d Cir. 1970) (same work from *Kalpakian* found to be infringed by verbatim copy); Herbert Rosenthal Jewelry Corp. v. Zale Corp., 323 F. Supp. 1234 (S.D.N.Y. 1971) (jewelled bee and turtle pins protectible).

[191] *See* Gates Rubber Co. v. Bando Chem. Indus., 9 F.3d 823, 838 (10th Cir. 1993); *M. Kramer Mfg. Co.*, 783 F.2d at 436; *Whelan Assocs.*, 797 F.2d at 1237; *Apple Computer*, 714 F.2d at 1253 (" '[I]f the same idea can be expressed in a plurality of totally different manners, a plurality of copyrights may result. . . .' " Merger occurs when "there are no or few other ways of expressing a particular idea.") (quoting Dymow v. Bolton, 11 F.2d 690, 691 (2d Cir. 1926)).

In *Paperback*, the court stated:

> If . . . the expression of an idea has elements that go beyond all functional elements of the idea itself, and beyond the obvious, and if there are numerous

idea and its alleged expression are truly inseparable, there can be no selectivity sufficient to satisfy the originality requirement.[192] If, on the other hand, an author has choices regarding the content or design of a work and imbues the work with more than a minimal amount of expression,[193] the court should not focus on copyrightability, but instead on the scope of protection. Thus, the better approach, that of the District of Columbia[194] and Second Circuits,[195] is that merger is relevant at the infringement stage as a limiting principle on the scope rather than on the existence of protection. When used at the infringement stage, merger can be applied sensitively to the facts before the court, permitting the court to ensure that the proper balance between protection and competition is preserved.[196] A work with minimal originality (a "thin" copyright in *Feist*'s lexicon)[197] may be protected only against verbatim or near verbatim copying—nevertheless important protection—while the creation of original works that genuinely build on the earlier, "thin" copyrighted work will not be impeded. Thus used, merger and other doctrines such as the fair use privilege, can be important, almost surgical tools to strike the appropriate balance in individual cases.[198]

By contrast, when merger is used at the copyrightability stage, if too much protection is granted in one case, it will be difficult for later courts (at least in the same circuit) to cut back on the protection for a given work even though in a later case the facts may

> other ways of expressing the non-copyrightable idea, then those elements of expression, if original and substantial, are copyrightable.

Paperback, 740 F. Supp. at 59.

This statement is awash with confusion between the copyright and patent laws (requiring that the expression be "beyond the obvious"), *id.*, and seemingly requires that a higher standard for copyright protection be met than that set forth by the Supreme Court in *Feist*. *See supra* text accompanying note 162.

[192] *See* Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003, 1012 (N.D. Tex. 1978) ("One can argue that inseparability of idea and expression is here an antinomy. The argument asks if the idea and the usage are not separable, what is the expression?").

[193] The mere fact that there are choices does not automatically mean that the choice made possesses originality. This is one of the mistakes made by the Third Circuit in *Whelan Associates*, discussed *infra* text accompanying notes 216-22.

[194] *E.g.*, *Atari Games Corp.*, 888 F.2d at 884-86 (*scénes á faire*).

[195] *E.g.*, Kregos v. Associated Press, 937 F.2d 700, 705 (2d Cir. 1991); Continental Casualty Co. v. Beardsley, 253 F.2d 702, 706 (2d Cir.), *cert. denied*, 358 U.S. 816 (1958).

[196] As Chief Judge Newman noted:

> Determining when the idea and its expression have merged is a task requiring considerable care: if the merger doctrine is applied too readily, arguably available alternative forms of expression will be precluded; if applied too sparingly, protection will be accorded to ideas.

Kregos, 937 F.2d at 705.

[197] *See Feist*, 499 U.S. at 349.

[198] *See infra* text accompanying notes 270-72.

warrant permitting a subsequent programmer greater leeway in copying. Conversely, when merger is used to deny protection altogether, later courts lose their ability to prevent verbatim copying by freeloaders. Unfortunately, in computer program cases the lower courts have elevated merger to be the principal criterion by which copyrightability is to be determined, rather than focusing on the statutory scheme Congress provided. Oddly, this elevation is also the result of a misinterpretation of a nineteenth century case, *Baker v. Selden*, decided ninety-six years before section 102(b) became part of the Copyright Act.

C. Baker v. Selden

The origin of the idea-expression dichotomy is frequently traced to the Supreme Court's 1880[199] decision in *Baker v. Selden.*[200] It is questionable whether the origin is correctly ascribed: the opinion never refers to ideas and was decided on the ground of lack of originality.

Plaintiff Selden had developed a system of double-entry bookkeeping, which he described through explanatory text and forms and published in a book for which he received a copyright registration. Defendant Baker wrote a book employing a similar system of bookkeeping, but, critically, he used substantially different forms. While certain columns were identical in both parties' works, the defendant claimed these columns did not originate with the plaintiff and were, moreover, required by state law.[201] Even though there was no appropriation of either the explanatory text or the forms, Selden claimed that the copyright in his book gave him the exclusive right to *use* the double-entry system of bookkeeping despite the lack of substantial similarity between the parties' forms.[202] The Supreme Court rejected Selden's claim of ownership of rights

[199] Although the date of this opinion is frequently given as 1879, that date is inaccurate. I have confirmed in the records of the Supreme Court that the correct date is 1880.

[200] 101 U.S. 99 (1880).

[201] *Compare* Baker v. Selden, 101 U.S. Records & Briefs 10-13 (1880) (argument for appellant) *with* 101 U.S. Records & Briefs 9 (argument for appellee).

[202] *See* Baker v. Selden, 101 U.S. Records & Briefs 6-7. The appellee argued:

> Ruled lines and headings in the abstract may be open to the common use of all in bookkeeping—but when they are so arranged as to bespeak by classification and condensation a new method of keeping accounts, they, too, are protected, because they then become something more than mere ruled lines and headings in the abstract, and convey useful knowledge in the concrete. . . . The appellee claims that these copyrights extend to and embrace a *system of bookkeeping*—a combination of lines as arranged as to suggest an improved method of classifying and condensing mercantile accounts. The lines are the symbols to convey the idea of the method or plan.

Id. (emphasis supplied).

in his system, writing, "the mere copyright of 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."[203]

The Court's reasoning is revealed in an earlier passage:

> To give to the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters-patent, not of copyright.[204]

In short, Selden was attempting to gain patent protection for his system through a copyright in a book, an effort the Court rightly rejected. Then, in dictum that has become (in)famous as the "use versus explanation" dichotomy, the Court added:

> The very object of publishing a book on science or the useful arts is to communicate to the world the useful knowledge which it contains. But this object would be frustrated if the knowledge could not be used without incurring the guilt of piracy of the book. And where the art it teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public; not given for the purpose of publication in other works explanatory of the art, but for the purpose of

[203] *Baker*, 101 U.S. at 107; *see* Perris v. Hexamer, 99 U.S. 674, 676 (1878).

[204] *Baker*, 101 U.S. at 102. Claims for patent protection by others did not, however, fare any better. *See* Munson v. Mayor of New York, 124 U.S. 601, 601 (1888) (finding invalid a patent for "new and useful improvements in preserving, filing and canceling bonds, coupons, certificates of stock, &c.," consisting of blank books). Other cases denying copyright in systems or plans include Crume v. Pacific Mut. Life Ins. Co., 140 F.2d 182 (7th Cir. 1944); Taylor Instrument Cos. v. Fawley-Brost Co., 139 F.2d 98 (7th Cir. 1943), *cert. denied*, 321 U.S. 785 (1944); Brief English Sys., Inc. v. Owen, 48 F.2d 555 (2d Cir.), *cert. denied*, 283 U.S. 858 (1931); Stone & McCarrick, Inc. v. Dugan Piano Co., 220 F. 837 (5th Cir. 1915) (manual on how to create advertisements did not prohibit piano company from using illustrations contained therein to create an advertisement); Amberg File & Index Co. v. Shea Smith & Co., 82 F. 314 (7th Cir. 1897) (index books held noncopyrightable); Drugtax, Inc. v. Systems Programming Corp., 147 U.S.P.Q. (BNA) 313, 315 (M.D. Pa. 1965) (While defendant copied "in considerable detail" plaintiff's idea, plan, and method for enabling pharmacists to supply their customers with information on the deductibility of their drugs, the court held that "[t]his . . . is not the test in determining infringement of a copyright."); Gaye v. Gillis, 167 F. Supp. 416 (D. Mass. 1958) (coupon book); Aldrich v. Remington Rand, Inc., 52 F. Supp. 732 (N.D. Tex. 1942); Burk v. Relief and Burial Assoc., 3 Haw. 388 (D. Haw. 1909); Burnell v. Chown, 69 F. 993 (C.C.N.D. Ohio 1895) (credit ratings); Griggs v. Perrin, 49 F. 15 (C.C.N.D.N.Y. 1892). An interesting (but nevertheless unsuccessful) attempt to circumvent the lack of copyright in systems was made in Seltzer v. Sunbrock, 22 F. Supp. 621, 627 (S.D. Cal. 1938), in which the court rejected plaintiff's claim that a description in its pamphlet of a system for conducting roller skating races was a "dramatic composition." *See* Seltzer v. Corem, 26 F. Supp. 892 (N.D. Ind. 1939).

practical application.[205]

This statement is dictum for two reasons: first, under the facts of *Baker*, the defendant Baker's work was *not* substantially similar to plaintiff Selden's; Baker could use the knowledge contained in Selden's book "without incurring the guilt of piracy" and did so without copying Selden's forms. In other words, Baker *did* practice the art contained in Selden's book without using Selden's "methods and diagrams." The centerpiece of the suit was Selden's claim that he owned a copyright in the bookkeeping *system* notwithstanding the lack of substantial similarity in the forms—the illustrated "methods and diagrams."[206] If there had been substantial similarity between the parties' forms, the case would have been a run-of-the-mill infringement suit.

The second and perhaps more important reason the "use versus explanation" statement is dictum is that the Court decided the case on an entirely different ground. The holding as announced by the Court was: "blank account-books are not the subject of copyright."[207] The Court's conclusion was thus that Selden's forms lacked originality, not that they contained expression which was merged with his bookkeeping system, as the lower courts' computer program opinions mistakenly assert.[208] Instead, there was no original expression to begin with. The holding announced by the Court—that there is no originality in blank forms—spawned a rich history of cases and Copyright Office regulations involving such forms,[209] the upshot of which is a conclusion that the *Baker v. Selden* "blank form rule" is nothing more than an application of the originality requirement. Those forms that possess the requisite "modicum of creativity" have been protected.[210]

[205] *Baker*, 101 U.S. at 103.

[206] The Supreme Court appears to have subsequently interpreted *Baker v. Selden* as standing for nothing more than a case where there was no substantial similarity and the claim was in the system notwithstanding the lack of similarity. *See Mazer*, 347 U.S. at 217 ("[I]n *Baker v. Selden*, the Court held that a copyrighted book on a peculiar system of bookkeeping was not infringed by a similar book using a similar plan which achieved similar results where the alleged infringer made a different arrangement of the columns and used different headings." (citation omitted)).

[207] *Baker*, 101 U.S. at 107.

[208] *See Computer Assocs.*, 982 F.2d at 704-05; *Lotus Dev. Corp.*, 49 F.3d at 817. The facts in *Baker v. Selden*—Baker's ability to illustrate Selden's system using nonsubstantially similar forms—demonstrate that there was no merger.

[209] *See* Patry, *supra* note 2, at 328-32.

[210] *Id.* at 332; *see Kregos*, 937 F.2d at 708-09. Application of the "blank form" originality "rule" to computer works has been inconsistent. The first case to do so, *Synercom*, 462 F. Supp. 1003, arose under the 1909 Copyright Act and involved complex facts. Data was contained in copyrighted manuals. This data was then manually written down on "input format" cards, which had the appearance of an 80-column punched card. Once the data was transferred to the input format cards, the data was given to operators who created

Baker v. Selden, when understood according to its facts and the holding announced by the Court, stands for the unremarkable conclusion that Selden's forms lacked the requisite originality. Since the forms were unprotected and Selden had not received a patent, he had no right to control their use. Accordingly, deification of *Baker v. Selden* in late twentieth century computer program cases is startlingly unjustified.

For example, in *Computer Associates*,[211] the Second Circuit regarded a computer program's sets of statements or instructions as roughly analogous to Selden's process of bookkeeping.[212] This equivalency, if consistently applied, would eliminate protection for any computer program, since every computer program, according to the statutory definition, must bring about a "certain result," and thus each is a "process" for bringing about that result. This is exactly what happened in *Lotus Development Corp.*,[213] where the First Circuit stated: "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)."[214]

punch cards containing the data in a form executable by a computer. Each input format was individually registered with the Copyright Office. Defendant did not copy the computer program code, but it did copy information from the manuals, and plaintiff alleged it also copied the input formats. Then-District Judge Higginbotham found the input formats conveyed information, but that the information was merged with the "idea" of the work. The continuing precedential value of *Synercom* is open to question, however, since in a recent case involving the same parties and many of the same input formats (but in a reverse posture as parties), the Fifth Circuit upheld a claim of copyrightability and infringement in the formats as a compilation. *See Engineering Dynamics*, 26 F.3d 1335. In the process, the Fifth Circuit pointedly disavowed the argument that it had endorsed *Synercom* in its opinion in Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987). *See Engineering Dynamics.*, 26 F.3d at 1341-42.

Other cases have confronted the blank form rule in connection with computer program elements. *See, e.g., Whelan Assocs.*, 797 F.2d at 1242-43 (protection for file structures); Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1040 (N.D. Cal. 1992) (gray background for computer terminal display found to be an "unprotectible blank form"), *aff'd*, 35 F.3d 1435 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995); CMAX/Cleveland, Inc. v. UCR, Inc., 804 F. Supp. 337, 355 (M.D. Ga. 1992) (file structures found not to be blank forms); *Digital Communications Assocs.*, 659 F. Supp. at 460-62 (finding status screen display was not a blank form).

[211] 982 F.2d 693.

[212] *Computer Assocs.*, 982 F.2d at 704-05. *Computer Associates* also misinterpreted *Baker* as holding that Selden's ledger sheets were unprotectible because they were "necessary incidents" to his process of accounting, and that any elements of a computer program that are "necessary incidents" to its function are therefore unprotected as well. *Id.* This conclusion fatally conflicts with the definition of "computer program," which *requires* a functional result. Obviously, the "set of statements or instructions" are necessary to that result. *See infra* text accompanying notes 236-37.

[213] 49 F.3d at 817.

[214] *Id.*

Whatever the merits of *Baker*'s dictum on use,[215] we now are construing a statute, a statute that extends protection to computer programs that "bring about a certain result." To construe *Baker* as prohibiting protection for the Lotus 1-2-3 menu command hierarchy because it assists in bringing about a certain result in a computer would be to deny protection for the very reason Congress granted protection.

VIII. Disarray in the Lower Courts

Regrettably, the lower courts have ignored the language and structure of the Act, and in that ignorance have formulated their own solutions to the scope of copyright in computer programs. This freelancing is best revealed by three opinions, *Whelan Associates v. Jaslow Dental Laboratory, Inc.*, *Computer Associates International, Inc. v. Altai, Inc.*, and *Lotus Development Corp. v. Borland International, Inc.*

A. Whelan Associates v. Jaslow Dental Laboratory, Inc.

Whelan Associates was a suit over a computer program designed to aid in the management of dental laboratories. There was no allegation of copying the literal code since the defendant's work was written in a different computer language for a different type of computer.[216] Nor was there an allegation that the defendant's work was a mere translation since the program had to be revised to

[215] *Cf. Digital Communications Assocs.*, 659 F. Supp. 449, which involved a "status screen" display. A status screen display contains an arrangement of command terms under various headings. Defendant argued that because the entire status screen was "used" as part of the operation of the computer program, protection was prohibited by *Baker v. Selden.* The court rejected the argument, writing:

> The defendants read into *Baker v. Seldin* [sic] a distinction between works to "be used" and works to "explain." As noted in *Mazer v. Stein*, "nothing in the copyright statute [supports] the argument that the intended use or use in industry of an article eligible for copyright bars or invalidates its registration. We do not read such a limitation into the copyright law." A work can be used and can explain; that part of the work which explains, if not necessary to the idea of the work, is copyrightable.

Id. at 459 (quoting *Mazer*, 347 U.S. at 218) (citations omitted).

Unfortunately, Judge O'Kelley didn't go far enough; there is also no bar to protecting a work in the form in which it is being used so long as it contains expression. Indeed, under Judge O'Kelley's view, no operating system program could be protected because an operating system is only "used"; it does not interact with the consumer. His approach is strongly reminiscent of dissenting CONTU Commissioner Hersey who argued that no protection should be granted for the "usable forms of computer programs." *See* CONTU Final Report, *supra* note 120, at 27. CONTU rejected this view, as has every other court that has addressed the issue, beginning with the landmark case of *Apple Computer*, 714 F.2d 1240.

[216] The original program was written in the EDL language for defendant's IBM Series/1 minicomputer. The infringing program was written in the BASIC language so that it could run on the popular IBM PC.

operate on the different computer. Instead, the complaint was that the defendant had copied nonliteral elements: the program's "structure, sequence, and organization."[217]

In finding that structure, sequence, and organization was a protectible element of the plaintiff's work, the Third Circuit relied on the following syllogism:

> 1. Plays are literary works.
>
> 2. Copyright in plays may extend beyond the literal words to detailed plots.
>
> 3. Computer programs are "literary works."
>
> 4. Therefore, copyright in computer programs extends beyond literal code to their detailed plot; i.e., structure, sequence, and organization and must be analyzed according to the same principles as are plays.[218]

As noted above,[219] the syllogism is faulty, although it has been accepted even by courts that otherwise disagree with *Whelan Associates.* The *Whelan Associates* court did create controversy in its formulation of the following rule for separating the idea from expression in computer programs:

> [T]he purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea. Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea.[220]

[217] For a chart comparing the structure, sequence, and organization of the parties' respective programs, see ANTHONY LAWRENCE CLAPES, SOFTWARE, COPYRIGHT & COMPETITION: THE "LOOK AND FEEL" OF THE LAW 106-08 (1989).

[218] *Whelan Assocs.*, 797 F.2d at 1234; *see Lotus Dev. Corp.*, 788 F. Supp. at 92 ("[T]he sequence, structure, and organization of the program's code, in addition to the text of the code itself, may be protected by copyright—under the same rationale that protects the detailed plot line and structure of a play in addition to the play's actual dialog.").

[219] *See supra* text accompanying notes 91-97.

[220] *Whelan Assocs.*, 797 F.2d at 1236 (citations and footnote omitted). *But see id.* at 1238 n.34:

> We do not mean to imply that the idea or purpose behind *every* utilitarian or functional work will be precisely what it accomplishes, and that structure and organization will therefore always be part of the expression of such works. The idea or purpose behind a utilitarian work may be to accomplish a certain function *in a certain way,* and the structure or function of a program might be essential to that task.

Id. (citation omitted). *See also Whelan Assocs.*, 609 F. Supp. at 1320. "The 'expression of the idea' in a software computer program is the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating, and producing useful information either on a screen, print-out or by audio communication." *Id.* Arguably, the Third Circuit did not follow its own rule in its analysis of the parties' works at issue in *Whelan Associates.*

This approach collapses the idea-expression dichotomy into a merger analysis. In so doing it omits a critical step in any copyright analysis—the determination of whether originality exists.[221] Material that is not necessary to the program's purpose or idea may be de minimis, or otherwise unoriginal, such as an algorithm. Thus, while the existence of other ways of achieving a particular result is evidence of a lack of merger, it is not conclusive on originality.

Whelan Associates presents not so much an analysis of the idea-expression dichotomy as a bypassing of the originality inquiry and a collapsing of the idea-expression dichotomy into a merger analysis. The merger analysis should take place at the infringement, not the copyrightability, stage.[222]

B. Computer Associates International, Inc. v. Altai, Inc.

In *Computer Associates International, Inc. v. Altai, Inc.*,[223] Second Circuit Judge George Pratt, sitting by designation as a trial judge, rejected *Whelan Associates.* In place of *Whelan Associates,* Judge Pratt adapted Judge Learned Hand's *Nichols* "abstractions" test,[224] thereby ironically falling into some of the same traps as the *Whelan Associates* court: using principles developed for plays to determine protection for computer programs.

Affirming Judge Pratt's decision, the Second Circuit went Judge Pratt one better, announcing the imposingly named "abstraction-filtration-comparison" test, a test that sounds—and acts—more like a water purification process than a copyright analysis. This test requires district courts to undertake a three-step procedure to determine which elements of the plaintiff's program are protectible and substantially similar to the defendant's program. As a general proposition,[225] district courts are directed to

> break down the allegedly infringed program into its constituent structural parts. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily

[221] Perhaps the *Whelan Associates* court omitted this step because there was no doubt that the program before it was sufficiently original.

[222] *See supra* text accompanying notes 194-98.

[223] 775 F. Supp. 544 (E.D.N.Y. 1991), *aff'd,* 982 F.2d 693 (2d Cir. 1992); *see* Micro Consulting, Inc. v. Zubeldia, 813 F. Supp. 1514, 1528 (W.D. Okla. 1990), *aff'd,* 959 F.2d 245 (10th Cir. 1992) (pre-*Computer Associates* case where the court rejected *Whelan Associates* and followed *Nichols*).

[224] *Nichols,* 45 F.2d at 121; *see supra* text accompanying notes 183-87.

[225] Judge Walker helpfully observed that changes in technology "can quickly outpace judicial decisionmaking. Thus, in cases where the technology in question does not allow for a literal application of the procedure we outline below, our opinion should not be read to foreclose the district courts of our circuit from utilizing a modified version." *Computer Assocs.,* 982 F.2d at 706.

> incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable material. Left with a kernel, or possible kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program. The result of this comparison will determine whether the protectable elements of the programs at issue are substantially similar so as to warrant a finding of infringement.[226]

The first step of this procedure is an adaptation of *Nichols*, and requires district judges to "dissect the allegedly copied program's structure and isolate each level of abstraction contained within it."[227] In a statement that must cause despair for overworked district judges in the Second Circuit, the court of appeals stated that they are to "retrace and map each of the [program] designer's steps—in the opposite order in which they were taken during the program's creation."[228] This obligation imposes on trial courts a task even more formidable than that faced by Ginger Rogers, who once quipped she had to do everything Fred Astaire did, only backwards and in high heels.

The second step is "successive filtration," undertaken after all of the program's abstraction levels have been identified. This step

> entails examining 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 and hence is non-protectable expression. . . . The structure of any given program may reflect some, all, or none of these considerations. Each case requires its own fact specific investigation.[229]

The third step, comparison, takes place once the court "has sifted out all elements of the allegedly infringed program which are 'ideas' or are dictated by efficiency or external factors, or taken

[226] *Id.*

[227] *Id.* at 707.

[228] *Id.* The probative value of retracing is uncertain. *Computer Associates* breezily asserts its necessity, but does not explain what is to be gained from the effort. Analysis of each step in a copyright owner's process of creation is not undertaken with other subject matter, including the play at issue in *Nichols.* Instead, courts rightly focus on the final product. In any event, the benefits to be derived from learning the rationale for individual choices made by a programmer may be outweighed by a tendency to lose the forest for the trees, to say nothing of the tremendous time and expense added.

[229] *Computer Assocs.*, 982 F.2d at 707.

from the public domain . . . ."[230] At this step, "there may remain a core of protectable expression. In terms of a work's copyright value, this is the golden nugget."[231] For a purported test of copyrightability and infringement of the nonliteral elements of a computer program, *Computer Associates* is surprisingly fixated on the literal elements of the program, losing the forest for the trees in the process, or, more specifically, losing the nature of computer programs as a *set* of statements or instructions.[232] Under *Computer Associates*, the whole is considerably less than the sum of its parts.[233]

This shrinking of the scope of protection is the result not only of the fragmenting of the computer program's set of statements or instructions into constituent elements (which are then ruthlessly dissected), but also of the court's treatment of computer programs' ultimate (unprotectible) function as "the composite result of interacting subroutines."[234] This composite result of interacting subroutines is, however, exactly how the Copyright Act defines originality in computer programs.

Of equal concern is the *Computer Associates* court's use of merger to deny protection to what are frequently the most creative components of a computer program. Citing the "use versus appli-

[230] *Id.* at 710.

[231] *Id.* The "nugget" passage appears to negate the court's earlier, helpful, statements that "[t]he functions of the modules in a program together with each module's relationships to other modules constitute the structure of the program," *id.* at 698 (quoting Steven R. Englund, Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 871 (1990)) (internal quotation marks omitted), and that "a program's structure includes its non-literal components such as . . . the . . . organization of inter-modular relationships . . . .", *id.* at 702, since that structure is first subject of the abstraction analysis and the "successive filtration" method. By subjecting each module to these two analytical processes, it is likely that the relationship between modules will be lost. In any event, the court's repeated references to "kernels" and "nuggets" suggest that it takes a segmented view of structure, rather than an organic one derived from the nature of programs as a "set" of statements or instructions. *Contra* John M. Walker, Jr., *Copyright Protection: Has Look and Feel Crashed?*, 11 CARDOZO ARTS & ENT. L.J. 721, 729-31 (1993) (Judge Walker offers a spirited defense of his opinion in *Computer Associates*, especially rejecting arguments that the opinion "eschewed the principle that the selection and organization of non-protectable material may itself be subject of copyright."). Of course, saying it isn't so doesn't mean that it isn't.

[232] It is perhaps telling that as in *Whelan Associates*, the *Computer Associates* court of appeals did not analyze the computer program before it according to its own test.

[233] By contrast, at least the Tenth Circuit, in adopting the *Computer Associates* test in a modified form, instructed the lower courts to "first compare [the works] in their entirety without filtering out the unprotected elements. . . . [A]n initial holistic comparison may reveal a pattern of copying that is not obvious when only certain components are examined." *Gates Rubber Co.*, 9 F.3d at 841. Unfortunately, this "holistic" approach is used only to establish whether the defendant copied, and not to determine copyrightability. *Gates Rubber Co.*, like *Computer Associates*, still loses the flow of computer programs by dissection. *But see Engineering Dynamics*, 26 F.3d at 1348 ("The ultimate focus . . . should be on the input formats and output reports taken as a whole.").

[234] *Computer Assocs.*, 982 F.2d at 705.

cation" language from *Baker v. Selden*,[235] the court of appeals stated "those elements of a computer program that are necessarily incidental to its function are . . . unprotectible."[236] Since the computer program's function was defined by the court as "the composite result of interacting subroutines,"[237] any number of otherwise protectible components could be viewed as incidental to that function.

The court's emphasis on efficiency is also troubling. Although recognizing that more than efficiency may go into the creation of a program,[238] the court made a number of unsupported conclusions that result in a significant lessening of protection for the class of computer programs. For example, the court states that

> In the context of computer program design, the concept of efficiency is akin to deriving the most concise logical proof or formulating the most succinct mathematical computation. Thus, the more efficient a set of modules are, the more closely they approximate the idea or process embodied in that particular aspect of the program's structure.[239]

This passage equates a technical concept (efficiency) with a legal one (the nonprotectibility of ideas or processes) without explaining why the equation is true. An important unarticulated premise is that efficiency (an undefined term) necessarily precludes expression (in the copyright sense). No evidence to support the premise is provided, however, and such an absolute correlation is highly suspect.[240] Constant innovation is the *sine qua non* of survival in the software field. At any given point the latest innovation may appear to be the most "efficient" solution, that is, until it is superseded by a newer program rendering the prior approach outdated.[241] Does the earlier solution cease to be an "idea"

[235] *See supra* text accompanying note 205.

[236] *Computer Assocs.*, 982 F.2d at 705; *see supra* note 212.

[237] *Computer Assocs.*, 982 F.2d at 705; *see supra* text accompanying note 234.

[238] *Computer Assocs.*, 982 F.2d at 708.

[239] *Id.* ("[E]fficiency concerns may so narrow the practical range of choice as to make only one or two forms of expression workable options.").

[240] The European Community rejected the efficiency argument in the course of enacting its 1991 software directive. *See* BRIDGET CZARNOTA & ROBERT HART, LEGAL PROTECTION OF COMPUTER PROGRAMS IN EUROPE—A GUIDE TO THE EC DIRECTIVE 44 (1991) ("Nor can a program be denied protection on the grounds that it represents the optimal solution to a given problem."). Judge Keeton, in an earlier opinion involving Lotus 1-2-3 and a different defendant, aptly described the evolutionary nature of the work's development and how others had built on the idea of an electronic spreadsheet program while expressing the idea in substantially different ways. *Paperback*, 740 F. Supp. at 65-66.

[241] Interestingly, following one of Judge Keeton's decisions in *Lotus Development Corp.*, Borland described Lotus's 1-2-3 menu command hierarchy as an "outdated user interface." Brief for Petitioner, *Lotus Dev. Corp.*, *supra* note 27, at 16.

at this later date because we now realize it is not, in fact, "the most efficient" solution?

In addition to being subjective, "efficiency" is a relative concept. Just as there is no one "idea" of a computer program, there is no one "most efficient" design, a fact that even the most vociferous opponents of copyright for computer programs acknowledge.[242] Sometimes efficiency is the result of incremental changes over past projects. Sometimes efficiency is the result of great creativity. The Copyright Act adjusts automatically to both scenarios—incremental improvements are granted limited protection, while improvements containing great originality receive more extensive protection in light of their creativity. By erecting an efficiency bar to all protection, courts are substituting a threshold that will deprive at least some deserving works of any protection. In any event, since Congress has not created an "efficiency" criterion for protection, the courts should not either.[243]

The *Computer Associates* approach suffers from a large number of ills. While the *Whelan Associates* rule is overly simplistic, the *Computer Associates* rule is overly complicated, and has been of practical assistance to only one type of person, the expert witness, whose services are essential for the detailed analysis of the programmer's creative process that the court's test requires.

C. Lotus Development Corp. v. Borland International, Inc.

1. The Trial Court

Since Borland conceded copying, its principal defense was that the Lotus 1-2-3 menu command hierarchy lacked the requisite originality. Borland argued there were eight "constraints"[244] on the menu command hierarchy's design.

[242] *See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2328 n.58 (1994):

> [E]fficiency must always be measured in terms of the programmer's goals. A design that is highly efficient in conserving memory may not be highly efficient in speed of execution. There is, as a consequence, no one "most efficient" design for a software program. Any well-designed program is as efficient in relation to its design goals as its developers have been able to make it, given their skill and the constraints under which they were operating.

It is interesting to note that the authoritative guide to the European Union's 1991 software directive states: "Nor can a program be denied protection on the grounds that it represents the optimal solution to a given problem." CZARNOTA & HART, *supra* note 240, at 44.

[243] *See* Diamond v. Diehr, 450 U.S. 175, 182 (1981) ("[I]n dealing with the patent laws, we have more than once cautioned that courts should not read into the patent laws limitations and conditions which the legislature has not expressed." (citation and internal quotation marks omitted)).

[244] *Lotus Dev. Corp.*, 831 F. Supp. at 212; *see id.* at 213 (" 'Constraint' implies a rule that must not be violated.").

> (1) Each menu command was chosen to tell the user the purpose of the menu command and its function.
> (2) Each menu was selected so that it had a different first letter from the other menu commands within the same menu.
> (3) Each menu was set up to have only about seven choices . . . .
> (4) The menus were structured so that words within the menu that dealt with similar functions were grouped together.
> (5) Executable operations that were likely to be frequently used were located near the top of the [menu] tree.
> (6) Menu commands within a menu were arranged from left to right in order of decreasing frequency of use.
> (7) Commands in submenus were grouped under the menu command to which they relate.
>
> Borland also proffered an eighth functional "constraint" of having each menu fit on one line of a computer screen. If this constraint were followed, each menu could have no more than 80 characters . . . .[245]

Judge Keeton found, however, that Lotus's menu command hierarchy acted in ways that rebutted each of Borland's arguments. He pointed to evidence of other spreadsheet programs using "vastly different menu trees,"[246] nonfunctional differences among those other spreadsheets,[247] and "the breadth of possible menu

[245] *Id.* at 212-13.

[246] *Id.* at 214; see also Judge Keeton's opinion in *Paperback*, 740 F. Supp. 37.

> Just as 1-2-3 expressed the idea of an electronic spreadsheet differently from VisiCalc, so did Microsoft's Excel. Originally written for the Apple Macintosh computer, it exploits the enhanced graphics capabilities of the Macintosh, as well as the mouse input device that is standard with the Macintosh. Excel has pull-down bar menus rather than a two-line moving-cursor menu, and a very different menu-command hierarchy.

Id. at 66.

[247] *See Lotus Dev. Corp.*, 831 F. Supp. at 214-15.

> Although differences in program functionality may explain some differences between menu trees, the differences in functionality cannot explain the breadth of differences among menu trees used in the various programs. For example, functional considerations do not explain why the "File" menu command is left-most in the Excel menu tree and fifth from the left in the 1-2-3 menu tree. Borland does not suggest or offer any evidence explaining how the (unspecified) differences in functionality among 1-2-3, Excel, and Quattro's native mode affect placement of the "File" command within the first menu. Similarly, 1-2-3 places the copy and move command in the first level of the menu tree. In Excel they are in the second level.
>
> The differences among menu trees in the various programs submitted are so large that they are, in a practical sense if not literally, incapable of enumeration. The broad scope of these differences cannot be explained in terms of differences in functionality. Indeed, Borland offers no evidence or argument providing a reasoned explanation (as opposed to an unsupported assertion) of how the magnitude of differences could be explained by any differences in functionality. I conclude that many of these differences are due to different choices about how to express to the user the available user choices about all the particular operations that the program can perform.

Id.

trees that may be achieved" in the same manner as 1-2-3.[248] He also found that "Borland provided no credible evidence explaining how functional considerations could completely control formation of the menu tree."[249] Judge Keeton's focus on the totality of the 1-2-3 menu command hierarchy was correct. While individual menu commands may be functional or are found in other electronic spreadsheet programs,[250] the same cannot be said for the totality of the 1-2-3 menu command hierarchy. Since Borland copied the menu command hierarchy in its totality, the only issue is whether the menu commands as a whole are protectible.

Judge Keeton also rejected Borland's argument that the hierarchy was an unprotectible system:

> [L]ike Lotus's menu tree, the protected expression of a compilation may be viewed as a system for accessing information. The purpose of a compilation is to communicate facts. The specific facts communicated, however, are not copyrightable. Copyright protects only the selection, arrangement and manner of presentation of the facts (to the extent that those elements meet the other requirements for copyrightability). *See* Feist Publications, Inc. v. Rural Tel. Serv. Co. . . . . The selection, arrangement, and manner of presentation in a compilation may provide the user with a *method* or systematic manner of accessing the (uncopyrightable) facts. Thus, copyright law protects only that part of a compilation that the reader actually *uses* for selection of facts that the reader wants to know. Nevertheless, the expressive aspects of a compilation remains copyrightable . . . . [T]he structure of the menu tree including its designated keys for invoking commands . . . may also be viewed . . . as a type of selection and arrangement of the executable operations in Lotus 1-2-3.[251]
>
> The menu tree expresses to the program user, in hierarchial fashion, the array of available operations. Even though the executable operations are not copyrightable, the menu tree is copyrightable because the (hierarchial) *arrangement* of the definition and identification of the executable operations contains expression.[252]

[248] *Id.* at 214; *see id.* at 217-18 (giving specific examples of the variety of expression available to Lotus).

[249] *Id.* at 213.

[250] Even here, Judge Keeton noted that "Borland's experts acknowledge that, given all of the various functional considerations, at least a limited range of choices remains for individual menu commands." *Id.*

[251] *Lotus Dev. Corp.*, 831 F. Supp. at 231.

[252] *Id.* at 211 (emphasis supplied); *see id.* at 216 (After citing *Feist*, the court stated: "the specific operations could be viewed as (uncopyrightable) facts and the menu tree as an

Borland did not deny that it had "copied the words and arrangement of the Lotus menu command hierarchy,"[253] and was, therefore, forced to argue that "the precise menu commands and structure are necessary to achieve . . . functional compatibility. Thus, the argument goes, the entire interface . . . is a functional system or 'idea' and is not copyrightable."[254] Judge Keeton rightly recognized the tautological nature of the argument:

> As applied to any case involving a [computer program], 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. The argument is an attempt to win by definition without focusing at any time on any substantive issue concerning the separation of idea and expression.[255]

While Judge Keeton's opinions provide a firm factual record, his failure to understand the relationship between section 102(a) and the section 101 subject matter definitions and his failure to analyze the definition of "computer program" renders his opinion incomplete.

2. Court of Appeals

Judge Keeton made extensive findings of fact, which were not reversed as clearly erroneous. The court of appeals appears to have ignored these extensive findings by regarding the issue of copyrightability as one of law, subject to de novo review, although it never explicitly said this was what it was doing. Given that the trial court had before it numerous examples of other spreadsheet products and testimony on the issue of whether there was a functional necessity for the composition of the Lotus 1-2-3 menu command hierarchy, the court of appeals's failure to discuss this testimony, or to address whether the trial court made clearly erroneous findings of fact based on that evidence, is troubling.[256] Even more troubling is the court of appeals's unique, grotesque misun-

arrangement (with textual labels) of the facts."); *Lotus Dev. Corp.*, 788 F. Supp. at 97 ("The selection of which, and how many commands to place in each menu level, and the organization of the successive menu levels into a coherent and intuitive menu structure, was an important and creative consideration in the development of 1-2-3.").

[253] *Lotus Dev. Corp.*, 49 F.3d at 812.

[254] *Lotus Dev. Corp.*, 799 F. Supp. at 216-17.

[255] *Id.* at 217.

[256] I do not suggest that the ultimate question of copyrightability is one for a jury. Judge Keeton found, at least tentatively, it was an issue for the court. *Lotus Dev. Corp.*, 788 F. Supp. at 96. I do suggest that where the court's decision on copyrightability is based, as it was in Judge Keeton's case, on extensive findings of fact, that an appellate court has at least an obligation to discuss why it views those facts differently. *Cf.* FED. R. CIV. P. 52(a) ("Find-

derstanding of how section 102(b) works, and its misapplication of *Baker v. Selden*, beginning with the conclusion that because 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 in section 102(b)."[257]

The First Circuit also resorted to a bizarre analogy to the buttons on a videocassette recorder in support of its conclusion that the menu command hierarchy is a "method of operation."[258] Lotus's compilation of 469 terms out of hundreds of thousands of possible terms, its creative and intuitive arrangement of them into fifty menus and submenus bears no resemblance to the de minimis "rewind," "stop," and "play" buttons on a VCR.

Because it found the entire set of menu commands was a "method of operation,"[259] the First Circuit failed to analyze the menu commands as a set of statements or instructions because it bypassed *any* analysis of originality. This "method of operation" approach is tautological and fatally inconsistent with the statute. It is tautological because it defines as the method of operation the very elements that comprise expression. The approach is fatally inconsistent with the statute because it denies protection to the set of statements or instructions on the very ground on which the statute grants protection—if the program brings about "a certain result." Under the First Circuit's analysis, *all* computer programs are unprotectible methods of operation because they bring about a certain result in the same manner the Lotus 1-2-3 menu command hierarchy does.[260]

ings of fact, whether based on oral or documentary evidence, shall not be set aside unless clearly erroneous.").

[257] *Lotus Dev. Corp.*, 49 F.3d at 817; *see supra* text accompanying notes 199-215.

[258] *Lotus Dev. Corp.*, 49 F.3d at 817.

[259] *Id.* at 815-18. At the same time, an equally curious interpretation of § 102(b) by Professor Nimmer must be noted:

> It would, then, be a misreading of Section 102(b) to interpret it to deny copyright protection to "the expression" of a work even if that work happens to consist of an "idea, procedure, process, etc." Thus, if a given "procedure" is reduced to written form, this will constitute a protectable work of authorship so as to preclude the unlicensed copying of "the expression" of the procedure, even if the procedure per se constitutes an unprotectable "idea."

1 NIMMER ON COPYRIGHT, *supra* note 144, § 2.03[D], at 2-35 (footnote omitted). To the extent this passage suggests that a method of operation or process becomes predictable merely by being fixed in a tangible form, it confuses fixation with originality. Unoriginal fixed works are not subject to protection. It is not fixation that results in protection, but originality. Under the Nimmers' approach, the unoriginal telephone white pages at issue in *Feist* would be protected by virtue of being reduced to written form. This is obviously incorrect.

[260] *See supra* note 64.

IX. Conclusion on *Lotus Development Corp.*

The First Circuit went so far astray because of its lack of analysis of the 1-2-3 menu command hierarchy under the statutory definition of "computer program," and because of its failure to understand the relationship between section 102(a) and section 102(b). Applying the approach set forth in part VI of this Article, one should analyze the Lotus 1-2-3 menu command hierarchy according to the statutory definition of a "computer program"—a set of statements or instructions to be used directly or indirectly in order to bring about a certain result.[261] One must decide whether the menu command hierarchy as a *set* of statements or instructions was "independently created by [Lotus] (as opposed to copied from other works)," and whether "it possesses at least some minimal degree of creativity."[262]

There is no dispute that the 1-2-3 menu command hierarchy was independently created. There should also be no dispute that as a *set* of commands, the hierarchy contains the requisite minimal degree of creativity: the judgment in selecting 469 specific menu commands and their arrangement into fifty menus and submenus, as well as the coordination of the hierarchy in a way that permits consumers to write macros evidences a high level of creativity.

Unlike the white pages in *Feist,* Lotus did not select all possible commands; it selected only 469. Borland's Quattro Pro version 1.0 menu command hierarchy had 5,215 commands.[263] Just by varying the ten menu commands that appear on 1-2-3's main menu, Judge Keeton developed over 250 possible combinations of commands.[264] The Second Circuit has upheld the copyrightability of a selection of only nine categories of pitching statistics under a similar theory.[265]

Significantly, Borland did not contest the following statement of material fact regarding the arrangement of the 469 menu commands into fifty menus and submenus:

> 32. The selection of which, and how many, commands to place in each menu level, and the organization of the successive menu

[261] *See supra* text accompanying note 139.

[262] *Feist,* 499 U.S. at 345.

[263] *Lotus Dev. Corp.*, 831 F. Supp. at 211.

[264] *Lotus Dev. Corp.*, 799 F. Supp. at 217.

[265] *See Kregos,* 937 F.2d 700, *criticized on this point in* Patry, *supra* note 2, at 197 n.331. The selections made by Lotus in its menu command hierarchy dwarf the minimal ones made in *Kregos. Kregos* is cited only to demonstrate that an influential court of appeals in copyright cases, in an opinion by an outstanding copyright jurist—Chief Judge Jon O. Newman—upheld originality for a work containing far less selection than is present with the 1-2-3 menu command hierarchy.

> levels into a coherent and intuitive menu structure, was an important and creative consideration in the development of 1-2-3. 1-2-3's creators, particularly Mr. Kapor, devoted substantial effort during the late stages of the program's development to selecting the words and structural organization for 1-2-3's menus. (Kapor Aff., ¶¶ 72-101.)[266]

This uncontested testimony is sufficient to uphold the copyrightability of the menu command hierarchy under the *Feist* standard.

In *Gnu Business Information Systems, Inc. v. Social Secretary, Ltd.*,[267] Judge Zagel applied a similar analysis rejecting the argument that plaintiff's program as a whole was unprotectible because individual parts may have been unprotectible:

> The defendants' analytical dissection, though relevant to defining the scope of GNU's copyright, ignores that copyrighted works have a certain synergy in that the sum of their unprotected elements may be a protectible whole. Brown Bag [Software v. Symantec Corp.], 960 F.2d [1465,] 1476 n.4 [(9th Cir.), *cert. denied*, 113 S. Ct. 198 (1992)]; Stillman v. Leo Burnett Co., Inc., 720 F. Supp. 1353, 1361 (N.D. Ill. 1989). Each individual subroutine in the code, for instance, may be unprotectible in that it is the most efficient method for obtaining the result desired or it is in the public domain, but the arrangement of the subroutines in the program as a whole may be protectible. *Stillman*, 720 F. Supp. at 1361; *see Feist*, 111 S. Ct. at 1296 (selection, coordination, or arrangement of uncopyrightable facts in an original way is entitled to copyright protection). Arguably then, GNU may have a copyright in the selection and arrangement of certain unprotected elements and the similar arrangement of similar subroutines in the Mastors program may be found to infringe the copyright in the Morrell program.[268]

This passage captures the essence of the approach taken in this article. The references to *Stillman* are significant because, like *Atari Games Corp. v. Oman*, *Stillman* involved an audiovisual work whose individual frames were unprotectible, but which was protected as a "series of related images." The citation to *Feist* and its application to computer programs is a bull's-eye.[269] Borland conceded that it copied the words and arrangement of menu com-

[266] *Lotus Dev. Corp.*, 788 F. Supp. at 97.

[267] 1994 Copyright L. Dec. (CCH) ¶ 27,199 (N.D. Ill. Nov. 9, 1993) (denying defendant's motion for summary judgment).

[268] *Id.* ¶ 27,199, at 27,100-01 (footnote omitted).

[269] *See* GI Corp. v. United States Elecs. Components Corp., 1994 Copyright L. Dec. (CCH) ¶ 27,329 (N.D. Ill. 1994) (finding, on defendant's motion for summary judgment

mand hierarchy—that is, Lotus's original expression as contained in a "set of statements or instructions . . . used . . . in a computer in order to bring about a certain result." In so doing, Borland appropriated a material portion of the 1-2-3 computer program and infringed Lotus's copyright.

The motivating force behind the court of appeals's majority opinion in *Lotus Development Corp.* appears to be a visceral reaction to Lotus's position that Borland could not copy Lotus's menu command hierarchy in order to permit consumers who had created macros to run on Lotus 1-2-3 to transport them to run on Borland's Quattro or Quattro Pro.

> 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. . . . We find this absurd. . . .
>
> 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.[270]

The compatibility concerns of the court of appeals are misplaced under the facts in *Lotus Development Corp.*, since the case does not involve a competitor who developed a superior product but is being sued by a copyright owner intent on preserving market share by misuse of the copyright laws. Microsoft developed its "Excel" spreadsheet program and eclipsed 1-2-3 and Borland in the market. Microsoft did so in a noninfringing way and without being sued by Lotus. Borland also developed a noninfringing product. But, unlike Microsoft, Borland did not have confidence it could gain a foothold in the market by relying on the virtues of its own product. Instead, Borland attempted to gain a market share by copying, verbatim, some of the most attractive features of Lotus's program and then effectively advertising to potential customers: "Buy our product. Our product has Lotus's features too," a kind of two-for-one deal.

that sets of code data in a consumer remote control device may possess sufficient creativity to be protected as a compilation).

[270] *Lotus Dev. Corp.*, 49 F.3d at 817-18.

Admittedly, it is exasperating when, in the computer industry, ease of customer use takes a back seat to the maximization of profits. But the lack of standardization in the computer industry is a marketing, not a legal, issue. Those companies whose products are the least "user-friendly" will pay the cost of such a decision in the marketplace. It is not up to the courts, through the copyright law, to force software companies to market products in a way the courts believe to be better for consumers.

At the same time, when the copyright law is being misused to squelch third parties who have created original computer programs that do not freeload off the efforts of others, doctrines such as the fair use privilege provide the courts with a safety valve enabling them to ensure that the ultimate purpose of copyright—promotion of the Progress of Science—is furthered rather than impeded.

For example, different concerns would be present if there were a limited number of ways of configuring a menu command hierarchy, if there were only a limited number of ways of achieving certain spreadsheet functions, or if Lotus had a dominant market share and Borland had copied a small number of commands as a necessary adjunct to creating an original menu command hierarchy. Under these circumstances, either the merger doctrine, applied at the infringement stage,[271] or the fair use privilege[272] could be used to fine tune a result taking into account important issues such as the extent of the parties' creativity, the purpose and extent of the defendant's copying, the harm to any potential market of plaintiff by that copying, and whether the public interest would be best served by extending protection. None of these concerns are present in *Lotus Development Corp. v. Borland International, Inc.*, which involves nonmerged material copied in its entirety for purely commercial purposes in an attempt to deprive the copyright owner of its market share.[273]

[271] *See supra* text accompanying notes 194-98.

[272] *See, e.g.*, Sega Enters., Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992). The Ninth Circuit found fair use in a video company's disassembly of a competitor's product in order to discover small amounts of computer code used by a computer game console to "accept" cartridges for play. The defendant had previously created its video games for play on other game consoles, and plaintiff was demanding exorbitant licensing fees. *See also* Eckes v. Card Prices Update, 736 F.2d 859, 864 (2d Cir. 1984) (suggesting that fair use may be relevant where a work becomes the "standard" in the field and subsequent compilers must refer to it).

[273] For these reasons, Judge Keeton rightly rejected Borland's fair use defense. *Lotus Dev. Corp.*, 831 F. Supp. at 240-44.