COMPUTER PROGRAMS, USER INTERFACES, AND SECTION 102(b) OF THE COPYRIGHT ACT OF 1976: A CRITIQUE OF *LOTUS V. PAPERBACK*

PAMELA SAMUELSON*

I
INTRODUCTION

Judges in computer software copyright cases have paid scant attention to a provision of the copyright statute intended to limit the scope of copyright protection[1] in accordance with the principles of the most venerated of American copyright cases, *Baker v. Selden*.[2] That provision is section 102(b) of the Copyright Act of 1976. It states: "In no event does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle or discovery."[3] That section 102(b) has been so little analyzed in software copyright cases is especially surprising because it was added to the copyright statute in part to ensure that

* Professor of Law, University of Pittsburgh School of Law.

This article is dedicated to Robert Kastenmeier in appreciation for his many contributions to U.S. copyright law and policy, and in particular, his foresight in recognizing the need to codify some of the limiting principles of copyright law, such as those that appear in 17 USC § 102(b) (1988). The author wishes to thank Mitchell D. Kapor and Professors Ralph Brown, Dennis Karjala, and Jerome Reichman for their insightful comments on an earlier draft of the article and her research assistant Linda Sharif for her thoughtful, thorough, and cheerful help in the preparation of this article.

[1] See, for example, *Whelan Associates, Inc. v Jaslow Dental Lab., Inc.*, 797 F2d 1222 (3d Cir 1986) (section 102(b) mentioned, but not incorporated into the court's infringement test or infringement analysis); *Manufacturers Technologies, Inc. v CAMS, Inc.*, 706 F Supp 984 (D Conn 1989) (section 102(b) only invoked to render certain user interface navigational conventions unprotectable by copyright law); *Broderbund Software, Inc. v Unison World, Inc.*, 648 F Supp 1127 (ND Cal 1986) (one "see also" reference to section 102(b), but no discussion of the provision in the court's analysis of the copyright issues).

[2] 101 US 99 (1879). The extent to which Congress intended to codify *Baker v Selden* by enacting 17 USC § 102(b) (1988) is the subject of some academic debate. See J. H. Reichman, *Computer Programs As Applied Scientific Know-How: Implications of Copyright Protection for Commercialized University Research*, 42 Vand L Rev 639, 693-95 n288 (1989). Courts interpreting section 102(b) have construed it as codifying substantial portions of *Baker v Selden*. See, for example, *NEC v Intel Corp.*, 10 USPQ2d (BNA) 1177 (ND Cal 1989), and *Signo Trading Intl v Gordon*, 535 F Supp 362, 365 (ND Cal 1981). The words of the provision, particularly those that refer to the unprotectability of processes, procedures, systems, and methods of operation, are derived from *Baker v Selden* and the many cases following it. See notes 70-71 and accompanying text.

[3] 17 USC § 102(b).

copyright protection for computer programs would not be construed too broadly.[4]

In the eleven years since the copyright statute was amended to state explicitly that computer programs could be protected by copyright law,[5] there have been numerous controversies over the proper application of copyright law to computer programs.[6] Especially controversial have been several cases involving user interfaces of computer programs.[7] A case that was widely

[4] The House and Senate Reports state quite clearly: "Some concern has been expressed lest copyright in computer programs should extend 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 in the program are not within the scope of copyright law." HR Rep No 1476, 94th Cong, 2d Sess 57 (1976), reprinted in 1976 USCCAN 5659, 5670; S Rep No 473, 94th Cong, 2d Sess 54 (1976). Professor Arthur Miller testified at hearings leading up to passage of the 1976 Act that without a provision of this sort in the statute, copyrights in computer programs might become the equivalent of patents for important elements of programs. This, he asserted, would stultify development of software. See Hearings before the Subcommittee on Patents, Trademarks, and Copyrights, of the Senate Committee on the Judiciary, 90th Cong, 1st Sess 197 (1967). See also Letter from Professor Arthur Miller to Salem Katsch, Esq., 6 (Oct 29, 1985) (expressing the view that section 102(b) codified *Baker v Selden*; also indicating that it was for patent law, not copyright, to protect the processes, systems, and methods of operations embodied in programs).

[5] In 1974 Congress enacted legislation to establish the National Commission on New Technological Uses of Copyrighted Works (CONTU). Pub L No 93-573, title II, 88 Stat 1873, 1873-74. Among the issues this commission addressed was whether computer programs should be protected by copyright law. See National Commission on New Technological Uses of Copyrighted Works, Final Report ch III (1979) ("CONTU Report"). Although a majority of the CONTU members regarded the 1976 Act as already having provided copyright protection to computer programs, the CONTU majority nevertheless recommended that Congress amend the copyright statute to make explicit that programs could be copyrighted, and to define the term "computer program" and add a provision permitting back-up copying and modifications to software. CONTU's proposed amendments to the copyright statute were added to An Act to Amend the Patent and Trademark Laws, HR 6943, 96th Cong, 2d Sess (1980), which became Pub L No 96-517, 94 Stat 3007 (1980), codified at 17 USC §§ 101, 117 (1980).

Notwithstanding the 1980 amendments, there were still a number of challenges to the copyrightability of certain kinds of computer programs after the passage of this act, most notably those involving operating system programs and microcode. See generally Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 Stan L Rev 1329 (1987); Pamela Samuelson, *CONTU Revisited: The Case against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 Duke L J 663.

[6] An overview of the many controversies over the application of copyright to computer programs can be found in Pamela Samuelson, *Reflections on the State of American Software Copyright Law and the Perils of Teaching It*, 13 Colum-VLA J L & Arts 61 (1988). For a more in-depth treatment of copyright issues raised by the protection of structural abstractions of computer programs and user interfaces, see Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan L Rev 1045 (1989); *LaST Frontier Conference Report on Copyright Protection of Computer Software*, 30 Jurimetrics J 15 (1989) ("*LaST Frontier Report*").

[7] Because the first several software user interface copyright cases did not involve major software firms (see, for example, cases cited in note 1), the cases were not much noticed in the software industry, even though some of them laid the conceptual groundwork for the later, more publicized user interface lawsuits by Lotus, Ashton-Tate, Apple, and Xerox. Until these lawsuits, the general view in the technical community seemed to be that as long as someone wrote his or her own code, it was legal to develop a competing application program with the same or very similar user interface. The apparent legality of computer hardware clones may have caused those in the industry to think, particularly in view of the interchangeability of hardware and software, that software clones would be lawful as well. Software clones are thought by some in the technical community to offer many of the same benefits to consumers that hardware clones had done: more product choices, price competition, and improved or additional features. See generally Pamela Samuelson, *Why the Look and Feel of Software User Interfaces Should Not Be Protected by Copyright Law*, 32 Communications of ACM 563

expected to yield a landmark ruling on one set of controversial user interface issues was *Lotus Development Corp. v. Paperback Software International.* In June of 1990, Judge Keeton issued a lengthy opinion in which he ruled that Lotus's copyright in its popular 1-2-3 spreadsheet program protected the user interface of that program, and that Paperback had infringed the Lotus copyright by copying a number of aspects of that interface.[8]

This article will argue that the court in *Paperback* missed an important opportunity to provide an updated interpretation of *Baker v. Selden*, the copyright principles it represents, the embodiment of these principles in section 102(b), and how these principles should be applied in copyright disputes involving user interfaces of computer programs. Amidst the myriad issues and defenses in the case (some of which may not have been strategically advisable to assert[9]), the court was unable to see the pertinence of some important copyright principles that, if heeded, might have caused the judge to view the issues in the case somewhat differently.[10]

This article will also argue that the court was so distracted by Paperback's argument that "nonliteral elements" of computer programs were "uncopyrightable" that it failed to engage in an inquiry as to whether the aspects of the Lotus interface that Paperback copied were elements of the Lotus 1-2-3 spreadsheet or macrocommand "system." Had such an inquiry been made, and these elements been determined to be part of a system, they would, under the principles of *Baker v. Selden*, its progeny, and section 102(b), have been just as unprotected by copyright law as were the ledger sheets that the Supreme Court ruled were unprotectable components of Selden's bookkeeping system.

This article will go on to argue that the *Paperback* opinion fails to offer persuasive reasons for its ruling that the Lotus interface was "expressive." It will also criticize the court's analysis of the functionality of the Lotus user interface and the copyright implications of this functionality. Copyright protection for functional writings, such as spreadsheet programs, is

(1989). See also Pamela Samuelson & Robert Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits*, 30 Jurimetrics J 121 (1989) (reporting on a survey of user interface designers).

The other major user interface case besides *Lotus v Paperback* is the lawsuit by Apple Computer against Microsoft Corporation and Hewlett-Packard Co., claiming exclusive rights in certain features of the graphical user interface that have come to be associated with Apple. Because of the more graphical character of the Apple interface, this lawsuit involves a somewhat different range of copyright issues than the *Lotus v Paperback* dispute. See Samuelson, 32 Communications of ACM at 563 (cited in this note). This Apple lawsuit is currently awaiting trial.

[8] 740 F Supp 37 (D Mass 1990).

[9] See notes 45-52 and accompanying text.

[10] The *Lotus v Paperback* lawsuit was settled out of court some months after issuance of the opinion that this article will discuss at some length. Under the terms of the settlement agreement, Paperback agreed not to appeal the district court's ruling, to take VP-Planner off the market, and to pay $500,000 to Lotus. Although Paperback still regards the court's ruling as erroneous, it decided that settling the dispute was in its best business interest. See Staff Report, *Lotus to Receive $500,000 in Copyright Case*, Wall St J B4 col 2 (Oct 18, 1990); *Lotus Settles Copyright Case*, NY Times D4 col 3 (Oct 18, 1990).

traditionally quite "thin."[11] Generally, only exact or near-exact copying will be found to infringe. Had the court used this approach to judge infringement in *Paperback*, and persuasively explained what it found to be expressive about Lotus's interface, *Paperback* might have been the landmark opinion for which many were hoping, one on which the law of copyrighting computer programs could have been firmly built.

As things stand now, *Paperback* seems to have stirred up more controversy than it settled. It has met with distinctly mixed reactions in the technical community, in part because people are uncertain about how broad or narrow the court's ruling really was.[12] Several articles critical of the court's analysis have already appeared in the law review literature.[13] And a recent ruling of the Ninth Circuit Court of Appeals that found no expressiveness in the arrangement of commands for a spreadsheet program is in direct conflict with the ruling in *Paperback*.[14] There is an opportunity, in a new Lotus lawsuit against one of its spreadsheet competitors,[15] for Judge Keeton, or perhaps the

[11] See, for example, *Continental Casualty Co. v Beardsley*, 253 F2d 702 (2d Cir 1958); *LaST Frontier Report*, 30 Jurimetrics J at 18-19 (cited in note 6).

[12] See sources cited in Pamela Samuelson, *How to Interpret the Lotus Decision (And How Not to)*, 33 Communications of ACM 27 (1990).

[13] See, for example, Ronald Abramson, *Why* Lotus-Paperback *Uses the Wrong Test and What the New Software Protection Legislation Should Look Like*, 7 Computer L 6 (Aug 1990); D. Lee Antton & Gary M. Hoffman, *Copyright Protection and Innovation: The Impact of* Lotus Development v Paperback Software, 7 Computer L 1 (Aug 1990); Richard H. Stern, *Legal Protection of Screen Displays and Other User Interfaces for Computers: A Problem in Balancing Incentives for Creation against Need for Free Access to the Utilitarian*, 14 Colum-VLA J L & Arts 283 (1990); Comment, Lotus Development Corp. v Paperback Software International: *Broad Copyright Protection For User Interfaces Ignores the Software Industry's Trend Toward Standardization*, 52 U Pitt L Rev 689 (1991) (authored by Gerard Lewis, Jr.); Comment, *Copyright Protection for Computer Languages: Creative Incentive or Technological Threat?*, 39 Emory L J 1293, 1330 (1990) (authored by Elizabeth G. Lowry).

[14] *Ashton-Tate Corp. v Ross*, 916 F2d 516 (9th Cir 1990) (command structure for user interface of spreadsheet program held to be "idea" under section 102(b)). The Ninth Circuit decision in *Ross* was decided after the *Lotus v Paperback* ruling, but does not cite to that opinion. The *Paperback* opinion did not cite to the trial court opinion in *Ross*, even though it was decided before the court's ruling in *Paperback*. See *Ashton-Tate Corp. v Ross*, 728 F Supp 597, 602 (ND Cal 1989), aff'd, 916 F2d 516, 521-22 (9th Cir 1990). *Ashton-Tate v Ross* is discussed in notes 152-157 and accompanying text.

In a previous article, the author observed that software copyright cases have tended to be either lengthy and elaborately flawed, or so cryptic as to provide little guidance on the reasoning that led to the courts' conclusions. See Samuelson, 13 Colum-VLA J L & Arts at 71-72 (cited in note 6), citing *Whelan*, 797 F2d 1222, as an example of the lengthy but flawed variety, and *Plains Cotton Cooperative Ass'n of Lubbock Texas v Goodpasture Computer Service, Inc.*, 807 F2d 1256 (5th Cir 1987) (a case factually and legally quite similar to *Whelan*, but which rejects its legal conclusion and reasoning) as an example of the too cryptic variety. *Paperback* now joins *Whelan* in the lengthy and elaborately flawed category, and *Ross* joins *Plains Cotton* in the too cryptic category. Of the two kinds of cases, the lengthy and elaborately flawed variety is the more worrisome because judges in subsequent cases may tend to equate length and elaborateness with sound analysis, which is sometimes not the case. Judges in subsequent cases may find it easier to follow a prior ruling based on a lengthy analysis of the issues than to dissect the lengthy analysis, locate its flaws, and construct the sounder analysis that should have been done.

[15] *Lotus Dev. Corp. v Borland Intl, Inc.*, Civ A No 90-11662-K (D Mass). Judge Keeton will be presiding in this case. A similar Lotus lawsuit against the Santa Cruz Organization has been settled out of court. See *Lotus Settles with Santa Cruz*, NY Times C4 col 2 (June 18, 1991).

First Circuit Court of Appeals, to bring about the clarification in the law that is so sorely needed by the software development community.[16]

II
THE *LOTUS V. PAPERBACK* DISPUTE

The facts presented by the *Paperback* case are simple and straightforward. Lotus Development Corporation owns a copyright in the very popular spreadsheet program, Lotus 1-2-3. In the process of preparing its own spreadsheet program (VP-Planner) to compete with the Lotus program, Paperback Software decided to copy several aspects of the Lotus 1-2-3 user interface, including its command structure. This was done to give consumers already familiar with the Lotus commands and macrocommand facility a lower-priced alternative product to purchase, one that could not only achieve the same functionality as Lotus 1-2-3, but also offered some other desirable functionalities that the Lotus product did not then have.[17]

Lotus then sued Paperback for copyright infringement. The complaint detailed a number of specific aspects of the Lotus interface that Paperback allegedly had wrongfully copied. These included the instruction, command, and menu language of the Lotus 1-2-3 interface, the "structure, sequence, and organization" ("SSO") of its screen displays and sequences of screen displays, the macrocommands and syntax of Lotus 1-2-3, and the overall "look and feel" of the interface.[18]

[16] See, for example, Computer Science and Telecommunications Board, *Intellectual Property Issues in Software* 4-6, 51-57 (Natl Research Council, 1991) ("NRC Report").

[17] There were a number of differences between the Lotus and Paperback user interfaces. The Paperback program had opening screen displays identifying the program as VP-Planner which were unlike Lotus's. In general, Paperback's screens were wider than Lotus's and allowed users to hide certain columns of data. Its help screens were organized differently than Lotus's. The menu bar of VP-Planner appeared at the bottom of the screen instead of the top, as Lotus had for 1-2-3 (although a user of VP-Planner could move the menu bar to the top of the screen, if the user preferred). Paperback's menu of commands also generally listed "help" as the first command in the set. For some submenus, VP-Planner had some additional commands that related to the data base capability VP-Planner provided that 1-2-3 did not. VP-Planner also used somewhat different wording for explanatory long prompts. See *Paperback*, 740 F Supp at 70.

The court nevertheless found Paperback's interface to be "strikingly [] similar" to the Lotus interface. Id. The court observed that because VP-Planner allowed users to move the menu bar from the bottom of the screen to the top and some of the VP-Planner screens displayed an array of commands identical to Lotus's, "a user could easily think 1-2-3 rather than VP-Planner was the program in use." Id.

It is worth noting that this statement reflects a trademark-like analysis of the case: Infringement should be found because the two works were confusingly similar, as if the problem was that Paperback had copied the trade dress of 1-2-3. Trade dress theory for protecting the "look and feel" of computer programs has been suggested by some commentators. See, for example, Richard A. Beutel, *Trade Dress Protection For the "Look and Feel" of Software: A New Source of Proprietary Rights Protection for the Software Industry?*, 5 Computer L 1 (Oct 1988).

One of the grounds on which Borland seeks to distinguish its dispute with Lotus from the Paperback dispute is that no one could mistake its Lotus-compatible alternative interface for the Lotus program because of distinctive characteristics of the Borland screens and their presentations of the commands. Borland's Redacted Memorandum of Law Regarding the Phasing of the Case, Civ A No 90-11662-K, at 11 (June 7, 1991) ("Borland Brief").

[18] See Complaint of Lotus Development Corp., *Lotus Dev. Corp. v Paperback Software, Inc.*, Civ A No 87-76-K (Jan 1987). The "look and feel" claim was among the more controversial of the Lotus

Paperback asserted a number of defenses to these claims. Its chief defense was that it had copied only "uncopyrightable" elements of the Lotus program.[19] Under the umbrella of this main defense, Paperback made several arguments. One was that only "literal" elements of computer programs (that is, source or object code) are protectable by program copyrights. Because user interfaces were "nonliteral" elements of programs, the copying of a user interface could not infringe a program copyright.[20] A second defense was that user interface screen displays are only protectable by copyright law if separately registered with the Copyright Office. Because Lotus had registered only the Lotus 1-2-3 program with the Copyright Office, Paperback argued that the court lacked jurisdiction over the copyright dispute pending before it.[21] A third defense was that the Lotus interface was a functional human-machine interface that was uncopyrightable under the "useful article" doctrine of copyright law.[22] A fourth defense was that Lotus was essentially claiming copyright protection for a computer language. Languages, Paperback asserted, were not copyrightable. Paperback insisted it had not infringed the copyright because it had copied only the language facility of Lotus 1-2-3.[23] Paperback's fifth argument was that it had been necessary for Paperback to copy various aspects of the Lotus interface in order to offer a "compatible" product to consumers. This seems to have been an argument

claims. The term is sometimes used in the technical community to describe the valuable functional behavior of a program that occurs when the user interacts with the program via the user interface. See, for example, *NRC Report* at 53 (cited in note 16). One reason for concern about the "look and feel" claim in the Lotus lawsuit among those who design user interfaces was uncertainty about its precise meaning. See Samuelson & Glushko, 30 Jurimetrics J at 127 (cited in note 7). As a matter of copyright law, there were a number of reasons to question whether the "look and feel" of programs could be protected by copyright law. See, for example, Samuelson, 32 Communications of ACM at 563 (cited in note 7). At trial, Lotus seems not to have emphasized the "look and feel" claim; *Paperback* states that the court did not find "look and feel" very helpful in resolving the copyright issues in the case. *Paperback*, 740 F Supp at 62.

[19] Two other Paperback defenses were laches and estoppel. Paperback asserted that Lotus had known, six months before VP-Planner appeared on the market, that it would look and work like 1-2-3, and yet had waited 14 ½ months after the Paperback product was released to raise any objections. Lotus also only made its objections known by initiating the lawsuit against Paperback. The court was persuaded that the delay in bringing suit was reasonable, citing cases that had found excusable delays to give the plaintiff time to evaluate the matter and prepare to bring suit. Id at 82. It may be, however, that Lotus was waiting to see what the Supreme Court would do with the appeal of the *Whelan* decision, for the Lotus lawsuits against Paperback and Mosaic were filed the day the Supreme Court denied certiorari in that case. For more discussion of the impact of the *Whelan* ruling on *Paperback*, see notes 30-52 and accompanying text.

[20] See notes 27-58 and accompanying text.

[21] *Paperback*, 740 F Supp at 79. Although the court "emphatically" rejected this defense as "frivolous" and based on a "word game" by the defense lawyers, it is worth observing that there was at least one precedent, *Digital Communications Assoc., Inc. v Softklone Distrib. Corp.*, 659 F Supp 449 (ND Ga 1987), that had ruled, after a thoughtful analysis on the relationship between computer programs and user interface screen displays, that such screen displays were separate works from the programs requiring separate copyright registrations. At the time Lotus filed the lawsuit against Paperback, the Copyright Office policy had not been clear on the subject of whether user interfaces needed to be separately registered or were covered by the underlying program copyright. See note 133 for further discussion of this issue.

[22] See notes 164-165 and accompanying text.

[23] See notes 95-113 and accompanying text.

that "idea" and "expression" had merged in the Lotus interface.[24] A sixth defense was that user interests in standardization of spreadsheet interfaces overrode the private interests of Lotus in the 1-2-3 interface.[25] Paperback also argued that on public policy grounds, copyright law should not be construed to extend protection to the elements of user interfaces for which Lotus was seeking protection because it would impede progress in the software industry.[26]

Judge Keeton wrote an extensive opinion rejecting each of Paperback's defenses. A close examination of this opinion suggests that certain defense strategies may have backfired and prevented the judge from engaging in some inquiries that would have proved more fruitful to the defense's objectives. By making more extreme arguments than necessary to advance their cause, the defense lawyers failed to focus the court's attention on the proper kind of copyright inquiry for such a case. As a consequence, a number of fundamental principles of copyright law that were quite relevant to the case went unanalyzed. Part III will demonstrate how one of Paperback's principal defenses got the copyright inquiry off on the wrong track.

III

GETTING DISTRACTED BY THE "UNCOPYRIGHTABILITY" OF "NONLITERAL" ELEMENTS DEFENSE

One of Paperback's principal defenses was that only "literal" elements of computer programs (that is, source and object code) were protectable by a copyright in a computer program. Paperback asserted that a user interface was a "nonliteral" element of a computer program, and as such, it was an "uncopyrightable" element of the program.[27] Whether it is appropriate to conceive of a user interface as a "literal" or "nonliteral" element of a computer program is a more complex intellectual problem than *Paperback* might suggest.[28] For purposes of this discussion, however, it will suffice to accept the characterization of the Lotus interface as a "nonliteral" element of

[24] See notes 101-112 and accompanying text.

[25] See note 94.

[26] Both Paperback and Lotus presented opinion evidence to the court concerning the effect that its ruling would have on the software industry. The court also took note of an article reporting the results of a survey of user interface designers that reflected strong opposition to the copyright "look and feel" lawsuits. See Samuelson & Glushko, 30 Jurimetrics J at 127-28 (cited in note 7). Although the court permitted the proferred evidence to be put into the record, it ultimately concluded that this kind of evidence was irrelevant to the copyright issues in the case. See *Paperback*, 740 F Supp at 73-77.

[27] The origins of the "uncopyrightable subject matter" characterization of the kind of defense raised in *Paperback* can be traced to *Baker v Selden*, 101 US 99 (1879). As note 29 explains, the more modern characterization of this kind of defense would be that the aspect of the work that the plaintiff was seeking to protect was outside the scope of the copyright. Nevertheless, cases continue to discuss whether certain aspects of copyrighted works are "copyrightable." See, for example, *Freedman v Grolier Enterprises, Inc.*, 179 USPQ (BNA) 476 (SDNY 1974) (discussing the "copyrightability" of a certain notation system for playing cards).

[28] See note 157 and accompanying text. It is worth noting here that notwithstanding the court's adoption of the "nonliteral" terminology, the opinion relies on "verbatim copying" to support the conclusion of infringement. See *Paperback*, 740 F Supp at 70.

the program and to concentrate, for the moment, on the effect the "uncopyrightability of nonliteral elements" defense had on the nature of the copyright discourse in the case.

One very important effect of this defense was to narrow dramatically the analytic focus of the case. The copyright discourse shifted from one about the proper scope of the Lotus copyright in the context of an infringement analysis to one about the "copyrightability" of "nonliteral" elements of computer programs.[29] The "nonliteral elements" defense was very risky because if the court could be persuaded that *any* nonliteral element of *any* computer program could properly be protected, the defense would founder. Several previous cases had ruled that some nonliteral elements of computer programs could be protected by copyright law, including the widely cited Third Circuit Court of Appeals decision, *Whelan Associates, Inc. v. Jaslow Dental Lab., Inc.*[30]

[29] The "nonliteral" elements defense affected the way in which the judge phased the trial for the case, as well as the "test" the court used in ruling on Lotus's claims. The first phase of the trial, as delineated in *Paperback*, was to determine "whether and to what extent plaintiff's computer spreadsheet program, Lotus 1-2-3, was copyrightable." 740 F Supp at 42. Phase two was to be a jury trial on issues of facts relating to whether the defendants had copied protected expression from 1-2-3, including whether defendants had copied expression from Lotus's source or object code. Id. The court resolved the entire case in the first phase proceeding. After ruling that the Lotus interface was a copyrightable component of the Lotus program, id at 68, the court decided that there was no genuine issue of fact requiring a phase two trial. The court observed that Paperback had conceded it had copied many elements of the Lotus interface, and these elements were, in the court's view, incontrovertibly substantial components of the program. Id at 68-70.

As part of its phase one proceeding, the court set forth three "elements" of what it called the "legal test for deciding copyrightability" in *Paperback*. Id at 59. The first "element," said to be derived from Judge Learned Hand's "patterns of abstraction" test in *Nichols v Universal Pictures Corp.*, 45 F2d 119, 121 (2d Cir 1930), was that "the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, *along the scale from the most generalized conception to the most particularized*, and choose some formulation—some conception or definition of the 'idea'—for the purpose of distinguishing between the idea and its expression." Id at 60 (emphasis in the original). The second element of the test was that "the decisionmaker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of *that* idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of that idea." Id at 61 (emphasis in the original). The third was "having identified elements of expression not essential to every expression of the idea, the decisionmaker must focus on whether those elements are a substantial part of the allegedly copyrightable 'work.' " The court indicated that this was to be judged on a qualitative, not just a quantitative, basis. Id.

This test is a highly idiosyncratic and erroneous analytic procedure for assessing the copyright issues raised by *Paperback*. A more appropriate way to frame the copyright issues in *Paperback* would have been to inquire, as part of a copyright infringement analysis, whether the aspects of the Lotus interface for which Lotus was seeking protection were properly considered within the scope of copyright protection available for the Lotus program. That is, were these aspects part of the protectable elements of "expression" of the program? See Paul Goldstein, *Copyright Principles, Law and Practice* ch 7 (Little Brown, 1989).

In the author's view, no "copyrightability" issue is present in a computer copyright case as long as the source and object code of the program meet the requirements of section 102(a) as an "original work[] of authorship fixed in any tangible medium of expression." See 17 USC § 102(a). Paperback was not arguing that there was no original expression in the Lotus program, that the program was not fixed enough to be copyrighted, or that the program was otherwise disqualified from being considered a protectable "work of authorship" under the statute. Only these arguments are now considered to be "copyrightability" issues. One of a number of reasons that *Baker v Selden* is in need of an updated interpretation is that its framing of the issue in these kinds of cases as a "copyrightable subject matter" issue is not in keeping with how copyright law is interpreted nowadays.

[30] 797 F2d 1222 (3d Cir 1986). *Whelan* has been cited with approval in several cases for the proposition that nonliteral elements of programs can be protected by copyright law. See, for

Therefore, a defense based on the unprotectability of nonliteral elements of programs was, as a strategic matter, almost suicidal.

Paperback made much the same argument in the Lotus case as Jaslow had in *Whelan*. Support for the proposition that copyright protection is available only for source and object code of computer programs is said to come from three sources: the statutory definition of computer programs,[31] the absence of legislative history indicating an intent to bring nonliteral elements of programs within the scope of copyright protection,[32] and judicial decisions denying copyright protection to certain nonliteral elements of programs.[33] In addition, the defendants in *Whelan* and *Paperback* argued that there was a need for a "bright line" standard so that software developers would know what they could lawfully copy from other programs.[34] A narrow scope of copyright protection for programs was also claimed to be desirable so as not to interfere with the kind of incremental development that characterizes software innovation.[35]

example, *Manufacturers Technologies*, 706 F Supp at 992; *Pearl Systems, Inc. v Competition Electronics, Inc.*, 8 USPQ2d (BNA) 1520, 1524 (SD Fla 1988); *Broderbund*, 648 F Supp at 1133.

[31] "Computer program" is defined in the copyright statute as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 USC § 101 (1988). The user interface of a program can accurately be described as among the "results" that can be generated when a program is executed in a computer. The user interface itself is not among the set of statements or instructions constituting a program that can be processed in the computer to bring about results. See, for example, Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 Jurimetrics J 33 (1987); Stern, 14 Colum-VLA J L & Arts at 350 (cited in note 13).

[32] There is no consensus among former CONTU Commissioners and staff members about whether CONTU thought that nonliteral elements of programs, such as their "structure, sequence, and organization," would be protectable by copyright. The Executive Director of the Commission staff and the chairman of the CONTU subcommittee on the copyrightability of computer programs think not. See Kenneth A. Liebman, Salem M. Katsch & David D. Leitch, *Back to Basics: A Critique of the Emerging Judicial Analysis of the Outer Limits of Computer Program "Expression,"* 2 Computer L 1 (Dec 1985) (discussing Commissioner Miller's and Director Levine's views); Arthur J. Levine, *Comment on* Bonito Boats *Follow-Up: The Supreme Court's Likely Rejection of Nonliteral Software Copyright Protection*, 6 Computer L 29 (July 1989). Two other CONTU Commissioners disagree. See E. Gabriel Perle, Christopher Meyer & Victor Siber, Bonito Boats *Redux*, 7 Computer L 1 (Feb 1990) (with an appendix on the now-deceased CONTU Commissioner Nimmer's views on CONTU's intent with regard to protection of nonliteral elements of programs). Both sides have found something in the CONTU Report to support their views on this matter. Id. (It is, however, worth noting that all of the examples of wrongful copying that CONTU discusses are "literal" copying examples. See CONTU at 22 (cited in note 5)).

Because of this nonconsensus, the CONTU Report is not an especially helpful source of information about the Commission's understanding of what it was recommending. There is little else in the legislative history of the 1980 software amendments to indicate congressional intent on this issue. See Samuelson, 1984 Duke L J at 666 (cited in note 5). *Paperback* indicates that CONTU had not explicitly addressed the central issue presented in the Lotus case. 740 F Supp at 50. It nonetheless quotes from Nimmer's views and merely notes the differing views of Miller and Levine. Id at 51.

[33] See, for example, *Synercom Technology, Inc. v University Computing Co.*, 462 F Supp 1003 (ND Tex 1978); *Plains Cotton*, 807 F2d 1256.

[34] See *Whelan*, 797 F2d at 1237-38; *Paperback*, 740 F Supp at 73.

[35] *Whelan* at 1237-38; see *Paperback* at 77-79. *Paperback* treats this as a "policy" argument rather than as a principle of copyright law applicable to functional works. See notes 123-127 and accompanying text for further discussion of this issue.

The Third Circuit already had rejected this defense in *Whelan*. Because nonliteral elements of other "literary works" (a statutory classification into which computer programs are said to fall) had long been protected by copyright law,[36] the court thought it appropriate for such elements to be protected in programs as well. It rejected the "bright line" standard argument as inconsistent with the congressional intent to have courts use the traditional idea/expression distinction,[37] and the incremental innovation argument as one that, if accepted, would provide too little incentive to invest in software development.[38] Consequently, it concluded that nonliteral elements of a program, such as its "structure, sequence, and organization," could be protected by copyright law.[39]

Although rejecting the "bright line" standard proposed by the defense, the Third Circuit in *Whelan* seems to have offered a highly protectionistic one in its stead: "[t]he purpose or function [of a program is] the work's idea, and everything that is not necessary to that purpose or function [is] part of the expression."[40] Necessity, the court declared, was to be tested by determining whether there are more than a small number of ways to achieve the function.[41] Although this test has been extensively criticized by commentators,[42] it has been used with some frequency in the software copyright case law,[43] for it has the apparent virtue of all "bright line" tests of providing judges faced with complicated legal arguments about factually complex phenomena with a simple basis for making distinctions that will resolve the case before them.[44] That the court in *Paperback* took the Third Circuit's shortcut through what

[36] *Whelan*, 797 F2d at 1234; *Paperback*, 740 F Supp at 51. Although *Paperback* acknowledges that the statutory definition of a "literary work" is far broader than "works of literature" (which the term would ordinarily connote), it nonetheless cites a string of cases involving works of literature and other artistic and fanciful works in support of the proposition that nonliteral elements of copyrighted works can be protected by copyright law. Id. As Part V will show, functional writings generally have a much narrower scope of copyright protection than literature, drama, and other artistic and fanciful works.

[37] The aspect of the CONTU Report that both *Whelan* and *Paperback* considered to be a reliable reflection of congressional intent is that which expresses confidence that the courts could draw the proper line between "idea" and "expression" in computer program cases. See CONTU Report at 18-23 (cited in note 5). Had CONTU and Congress intended for the line between copyrightable and uncopyrightable elements of programs to be drawn between their literal and nonliteral elements, the courts reasoned that they would have said so. *Whelan*, 797 F2d at 1241; *Paperback*, 740 F Supp at 54, 73.

[38] *Whelan*, 797 F2d at 1237-38. For similar reasoning, see *Paperback*, 740 F Supp at 75-76.

[39] *Whelan*, 797 F2d at 1240.

[40] Id at 1236.

[41] Id.

[42] See, for example, Karjala, 28 Jurimetrics J 33 (cited in note 31); Menell, 41 Stan L Rev 1045 (cited in note 6); *LaST Frontier Report*, 30 Jurimetrics J at 90 (cited in note 6).

[43] See, for example, *Broderbund*, 648 F Supp at 1133; *Pearl Systems*, 8 USPQ2d (BNA) 1520 (both applying the *Whelan* test).

[44] *Paperback* indicates that despite their superficial appeal, "bright line" rules often result in injustice. To reach just results, the opinion states, it is necessary to carefully weigh the facts and circumstances of each case. 740 F Supp at 73. Yet, by adopting the *Whelan* approach, the court in *Paperback* seems to have unconsciously slipped into the error of a "bright line" standard, one that is inconsistent with copyright statute and case law, as will be demonstrated further in Parts IV and V.

would otherwise have been a dense thicket of copyright issues is evident from a close reading of the opinion.

In *Paperback*,[45] the court identified "the idea" of the Lotus program in terms of its general purpose or function, that is, as a "computer spreadsheet program."[46] This was, of course, unprotectable by copyright law.[47] More particular details of the program, such as the rotated "L" creating the spreadsheet grid, the "/" key for invoking the menu of commands, and the command structure, were treated, in accordance with the *Whelan* test, as presumptively expressive. Therefore, unless the court found them to be "necessary" elements of a spreadsheet program, they would be established as copyrightable expressive elements.[48] Although recognizing that there were some spreadsheet programs that did not use the rotated "L" as a spreadsheet grid or the "/" to invoke the menu of commands, the court in *Paperback* concluded that there was only a limited number of options for accomplishing the functions these details served.[49] The court thus concluded that these were necessary elements of spreadsheet programs, and consequently, were instances in which idea and expression had "merged," and what would otherwise have been "expression" had become "idea."[50]

The court further concluded that other elements of the Lotus 1-2-3 interface, particularly its command structure and its mode of presentation of the commands, were not necessary for accomplishing spreadsheet program functions. In support of this conclusion, the court relied on the fact that it was possible to design a spreadsheet program with a different command structure and mode of presentation than that used by Lotus, which was evident from examination of the interfaces of a number of other spreadsheet programs that were different from Lotus'.[51] Having determined that these

[45] See note 29 for the three elements to *Paperback*'s legal test for copyrightability.

[46] *Paperback*, 740 F Supp at 42.

[47] Id at 65.

[48] Id at 66-67. Under the court's test, finding particular details to be "expressive" would not automatically mean that copyright infringement would be found, but such a finding that would mean the court would have to proceed to the third step of determining if the details were substantial elements of the work. See note 29.

[49] *Paperback*, 740 F Supp at 66-67. Stern questions the factual basis of certain of the court's assertions that there were a limited number of ways to achieve these functions. See, for example, Stern, 14 Colum-VLA J L & Arts at 335 (cited in note 13). A better copyright analysis of these details of the Lotus interface, as with Paperback's use of "+" to represent addition and "-" to represent subtraction, would have been that these elements, because so commonly found in spreadsheet programs, had become conventional elements of such programs, which should be treated in the same manner *scènes à faire* typically are in copyright law—as unprotectable by copyright. The court in *Paperback* seems to have avoided this way of analyzing the copyright status of these elements because of the deep disfavor with which it viewed Paperback's "standardization" defense. See note 94.

[50] *Paperback*, 740 F Supp at 66-67.

[51] The court listed seven spreadsheet programs that had different menu structures from Lotus 1-2-3, but also said that the idea of a menu structure for a spreadsheet program "could be expressed in a great many if not literally unlimited number of ways." Id at 67-68. For similar statements, see *Whelan*, 797 F2d at 1239. What does not seem to have occurred to the court in either case is that the different interfaces of other programs might reflect use of different "systems" or "methods of operation," rather than differences in expression.

How unwilling the court in *Paperback* was to take seriously any "necessity" defense when a commercially valuable element of the Lotus program was at stake is demonstrated by the dicta in its aspects of the Lotus interface were expression, the remaining question the court addressed was whether they were substantial components of the Lotus program, to which the court thought it "incontrovertible" that the answer must be yes.[52]

The principal criticism that has been leveled at the *Whelan* test is that it takes an overly narrow view of what copyright law considers to be "idea."[53] On its face, the test begins with the presumption that there is only one idea to be found in every computer program, and that all else in the work is expression unless a necessity test takes it out of the expression category and propels it into the idea category.[54] The *Whelan* test fails to take account of the fact that the term "idea" in copyright parlance is not confined in its meaning to "abstract generalized conceptions," such as the general purpose or function of a program; rather, it is a metaphor used in copyright law to describe the unprotectable elements in a copyrighted work.[55]

More specifically, the *Whelan* test fails to consider the full text of section 102(b), which indicates that such things as processes, procedures, systems, and methods of operation are unprotectable by copyright law even when

analysis of Paperback's "compatibility" defense. Paperback had asserted that unless it employed the same menu structure as Lotus (to make its product "compatible" with Lotus), it could not hope to achieve commercial success in the marketplace. While the court was persuaded that Excel's success in the spreadsheet market showed that it was not necessary to copy the Lotus command structure in order to be commercially successful, the court went on to say:

> [E]ven if VP-Planner otherwise would have been a commercial failure, and even if no other technological ways of achieving macro and menu compatibility existed, the desire to achieve "compatibility" or "standardization" cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual "work."

Paperback, 740 F Supp at 69. This statement suggests that the court would find it difficult to accept "idea/expression merger" for any valuable feature of the Lotus program, although the opinion elsewhere recognizes merger as standard copyright doctrine. Id at 59. The statement also seems to reflect a willingness to let Lotus enjoy a complete monopoly in the electronic spreadsheet market, a result hardly in keeping with CONTU's assurances that copyright protection for programs would not lead to monopolization of the market for program products or deter entry of competitors. See CONTU Report at 23-24 (cited in note 5).

[52] *Paperback*, 740 F Supp at 68. The court stated, in support of this proposition, that the Lotus interface was "its most unique element, and is the aspect that has made 1-2-3 so popular." Id. Further proof of its substantiality was said to be that the defendants had bothered to copy it. Id.

[53] See, for example, *LaST Frontier Report*, 30 Jurimetrics J at 20 (cited in note 6).

[54] One of many reasons why this is an inappropriate approach is that it will contribute to confusion about what is protectable by copyright in a program and what is patentable. The general purpose or function of a program is probably no more patentable than it is copyrightable. A particular way of achieving a program function, however, may now be patentable. See note 113. Only a cursory study of patents is required to discern that patent specifications routinely describe the different ways that the general function(s) of the invention had been accomplished in the prior art as a basis for establishing the novelty of the claimed inventive new way to do it. Thus, the existence of other ways of doing something is as likely to indicate that there are other patentable methods for achieving the same function as it is to indicate that there are other copyrightable nonliteral expressions of how to do it. The *Paperback* decision, like the *Whelan* decision before it, contributes to the confusion about what is protectable by copyright and patent law in a program, instead of helping to resolve it.

[55] *Paperback* seems at one point to recognize this, for it states that the idea/expression distinction "embraces also the process-expression, method-expression, and useful-expressive distinctions." 740 F Supp at 53. But the court fails to carry through with this kind of analysis.

embodied in the text of a copyrighted work.[56] The *Whelan* test also ignores the legislative history of the 1976 Act indicating that section 102(b) was added to the statute in part to ensure that copyright protection for computer programs would not be construed too broadly, that is, to ensure that such things as the "methods" and "processes" of "a program" would not be protected by copyright law.[57] The text of section 102(b), as well as the case law properly interpreting it, demonstrates unequivocally that not all "nonliteral" elements of a program should be treated equally under copyright law, as they are under the *Whelan* test, and not all should be presumed to be expression. Because there may well be more than a few methods or systems of achieving some general purpose, the mere existence of alternatives does not demonstrate that a nonliteral aspect of a computer program is "expressive."[58]

Another reason that the *Whelan* test is unsuitable as a test for copyright infringement for computer programs is that programs are often assemblages of components that could be packaged separately instead of being combined. A spreadsheet program, for example, might include a calculation component, a graphing component, and a data base component. If the *Whelan* test is applied to a software package combining these functions into one program, "the idea" in the program will be an electronic spreadsheet, and the three components, because they are more specific details, would be treated as presumptively expressive. If, however, the three components were packaged as separate programs, graphing would now be "the idea" of the graphing component, and only lower level details would be presumptively expressive (even though some of them may well be methods or processes). This demonstrates how much of a "word game" the *Whelan* test can be.

The proper inquiry in copyright cases involving computer programs must be much broader than the highly protectionist *Whelan* "bright-line" test permits. The remainder of this article will set forth the nature of the inquiry the court might have made had it not been distracted by Paperback's nonliteral elements defense. Part IV shows that a proper copyright inquiry in the case would have addressed two questions: whether any of the aspects of the Lotus interface that Paperback had copied—most importantly, the Lotus command structure—were constituent elements of a system for managing

[56] Section 102(b) is duly quoted in a background section of *Paperback*. Id at 49. Although occasionally referred to in a cursory manner thereafter, neither this section nor its contents (except that referring to the unprotectability of "ideas") is discussed in the subsection of the opinion on functionality issues, id at 54-58, in the subsection describing the legal test for copyrightability used in the case, id at 59-62, or in the sections that analyze the "copyrightability" of various elements of the Lotus interface and Paperback's copying of them, id at 63-70.

[57] In a background section of the opinion, id at 49, *Paperback* quoted the relevant passage from the House Report (see note 4), but made no effort to give it content. The terms "methods" and "processes" are emphasized in the text because the House Report's use of the plural expression demonstrates quite clearly that Congress thought there would be more than one of them per program.

[58] *Baker v Selden*, discussed in notes 63-92 and accompanying text, is one example of this principle. See also *Bibbero Systems, Inc. v Colwell Systems, Inc.*, 893 F2d 1104 (9th Cir 1990), discussed in note 142 and accompanying text.

spreadsheet functions or for constructing macros, and whether some copying of these elements was required to permit others to use or "express" the same system. Part V discusses the kind of inquiry the court should have made about what was "expressive" in the Lotus interface and whether any "expressive" elements were appropriated by Paperback.

IV

WAS THE LOTUS COMMAND STRUCTURE PART OF THE LOTUS SPREADSHEET SYSTEM?

Paperback discusses the "idea/expression" distinction at some length[59] but contains no analysis of whether any aspects of the Lotus interface might have been part of an unprotectable "system" for managing spreadsheet functions or for constructing macrocommands even though there are statements in the opinion strongly suggesting that this was the case.[60] Section 102(b) indicates that "systems" are as unprotectable as abstract ideas, in harmony with a long line of copyright cases that go virtually unmentioned in *Paperback*.[61] The "venerable" *Baker v. Selden* case is among the numerous precedents holding that an arrangement of words that is a constituent part of a system is not within the scope of copyright protection for the work, no matter how valuable or innovative the arrangement might be.[62] In addition to reviewing *Baker v.*

[59] See *Paperback*, 740 F Supp at 53-61, 65-68. For discussion of the court's "useful/expressive" distinction, see id at 53-58.

[60] The strongest of these statements can be found in id at 65, discussed in notes 101-103 and accompanying text. See also, id at 67 (comparing the "menu command system" of Visicalc with that of Lotus); id at 78 (referring to use of "/" to invoke the "menu command system" and Lotus's "macrocommand facility" and "command facility").

[61] See, for example, *Affiliated Enterprises, Inc. v Gruber*, 86 F2d 958 (1st Cir 1936) (promotional system not protectable by copyright); *Brief English Systems, Inc. v Owen*, 48 F2d 555 (2d Cir 1931) (shorthand system not protectable by copyright on booklets about it); *Chautauqua School of Nursing v National School of Nursing*, 238 F 151 (2d Cir 1916) (twelve-step hypodermic injection procedure not protectable by copyright in lecture); *Long v Jordan*, 29 F Supp 287 (ND Cal 1939) (old age pension system not protectable by copyright on pamphlet); *Burk v Johnson*, 146 F 209 (8th Cir 1906) (system for organizing mutual burial associations not protected by copyright in pamphlet); *Amberg File & Index Co. v Shea Smith & Co.*, 82 F 314 (7th Cir 1897) (indexing system not copyrightable); *Griggs v Perrin*, 49 F 15 (2d Cir 1892) (stenography system not protected by copyright in book); *Arica Institute, Inc. v Palmer*, 1991 Copyright L Rptr (CCH) ¶ 26,712 (SDNY 1991) (spiritual system not protectable by copyright in training manuals); *Kepner-Tregoe, Inc. v Carabio*, 203 USPQ (BNA) 124 (ED Mich 1979) (system for teaching problem-solving techniques not protectable by copyright); *Freedman v Grolier Enterprises*, 179 USPQ (BNA) 476 (notation system for playing cards not protectable by copyright); *Aldrich v Remington Rand*, 52 F Supp 732 (ND Tex 1942) (tax recording system not protectable by copyright); *Muller v Triborough Bridge Authority*, 43 F Supp 298 (SDNY 1942) (traffic separation system not within scope of copyright on drawings); *Seltser v Sunbrock*, 22 F Supp 621 (SD Cal 1938) (roller derby system not protectable by copyright in book); *Stone & McCarrick, Inc. v Dugan Piano Co.*, 210 F 399 (ED La 1914) (system of salesmanship not protectable by copyright on instruction manual); *Simms v Stanton*, 75 F 6 (ND Cal 1896) (physiognomy classification systems not protectable by copyrights in books). See also *Healthcare Affiliated Services, Inc. v Lippany*, 701 F Supp 1142 (WD Pa 1988) (use of similar systems and methods in computer program not copyright infringement); *Fishing Concepts, Inc. v Ross*, 226 USPQ (BNA) 692 (D Minn 1985) (processes in computer program not protectable by copyright); *Midway Mfg. v Bandai-America, Inc.*, 546 F Supp 125 (D NJ 1982) (game rules not protectable by copyright).

[62] A number of the "system" decisions (cited in note 61) involve arrangements of words as elements of the system. *Arica Institute*, 1991 Copyright L Rptr (CCH) ¶ 26,712, is a recent example of

Selden as a representative of the word-arrangement-as-system-elements cases, this section of the article suggests that, properly construed, *Paperback*'s "language" defense was a section 102(b) "system" defense that the court should have taken more seriously.

A. *Baker v. Selden*: The Arrangement of Words as Part of a System

Baker v. Selden is an important precedent to consider in assessing the *Lotus v. Paperback* dispute, not only because it is the "venerable" Supreme Court decision that ruled that constituent elements of systems embodied in copyrighted works are not protected by the copyright,[63] but also because it seems to be the only prior copyright case to have involved a claim of copyright infringement based on spreadsheet similarities.[64] It is therefore somewhat surprising how little attention *Paperback* gives to *Baker v. Selden*. That case is discussed in two sentences in the middle of a long paragraph in a part of the opinion remote from the analysis of the merits of Lotus's claim and of Paperback's defenses.[65] Although *Paperback* makes a passing reference comparing the Lotus electronic spreadsheet to paper spreadsheets,[66] the larger similarities between the two cases, in terms of their facts, the parties' legal contentions, and the Court's ruling seem to have escaped the *Paperback* court's attention.

Paperback begins its brief discussion of *Baker v. Selden* by referring to it as a "seminal case"[67] that held "that the text of a book describing a special method of double-entry accounting on paper spreadsheets . . . was copyrightable *expression*, but that the . . . *idea* of this particular kind of double-entry bookkeeping, was not."[68] The Supreme Court's statement of its holding in the case was importantly different from the district court's description of it in *Paperback*. The Court actually said that Selden's copyright protected his "explanation" of the accounting method, but not the useful

a case in which the arrangement of words was considered to be outside the scope of copyright because of its role as an element of a system. Palmer had written a book discussing the Arica philosophy, and reproduced diagrams closely resembling those in Arica's manuals. Arica sued for copyright infringement. Palmer defended by asserting that what she had copied were elements of Arica's system for curing ego fixation problems, which were illustrated by the diagrams consisting of nine-pointed stars surrounded by a circle. Each point of the stars was inscribed with a word or phrase that symbolized a component of the system. The ordering of the components was also part of the system. The court stated: "The copyright laws do not confer a monopoly on Arica in the method of describing a particular and interrelated set of characteristics or traits." Id at ¶ 24,159. See also note 88 concerning the irrelevance of the value or degree of innovation of the system in judging its status under copyright law.

[63] CONTU described *Baker v Selden* as a "venerable case." CONTU was confident that it and cases like it would provide guidance to the courts in attempting to distinguish what in a program should be regarded as idea or expression. CONTU Report at 18 (cited in note 5).

[64] See notes 73-74 and accompanying text.

[65] See *Paperback*, 740 F Supp at 53-54. The "legal test for copyrightability" discussion is found in the opinion, id at 59-62, and the discussion of the expressive elements of the Lotus interface is found at 65-68.

[66] Id at 63.

[67] Id at 54.

[68] Id (emphasis in the original).

"art" (that is, the bookkeeping method or system) explained in the book.[69] This correct formulation of the rule of *Baker v. Selden* makes clear something that is sometimes forgotten about the case: *Baker v. Selden* is fundamentally a case about the unprotectability of the functional content of written works, and the right of others to copy that content to make use of it.[70] To speak of the case as concerning only the unprotectability of abstract ideas oversimplifies the Court's ruling.[71]

A more serious mischaracterization of *Baker v. Selden* appears in the next sentence of *Paperback*: "[t]he Court thus concluded that Baker did not

[69] *Baker v Selden*, 101 US at 105.

[70] *Baker v Selden* has truly been a "seminal case," for from it has grown a number of important doctrines of American copyright law: (1) that the scope of protection for writings embodying functional content is quite narrow, for the functional content is not protectable by copyright (see, for example, *Beardsley*, 253 F2d at 702; *Kepner-Tregoe*, 203 USPQ (BNA) at 124); (2) that constructing a useful article depicted in a copyrighted work does not infringe the copyright (see, for example, *Muller*, 43 F Supp 298); (3) that blank forms are not copyrightable (see, for example, *Bibbero*, 893 F2d 1104); (4) that when there are significant constraints on the manner in which an idea can be expressed, even using the same expression will not be infringing as an instance in which idea and expression are said to be merged (see, for example, *Morrissey v Proctor & Gamble*, 379 F2d 675 (1st Cir 1967)); (5) that when useful elements of a copyrighted work must be copied in order to be used by others, no copyright infringement should be found (see, for example, *Kepner-Tregoe*, 203 USPQ (BNA) 124); and (6) that which is within the subject matter of utility patent law is outside the subject matter of copyright, and that it would be a fraud on the public to give copyright protection to that which has not satisfied the standards and procedures required by the patent system or to that which is the subject of an expired patent (see, for example, *Brief English Systems*, 48 F2d at 555; *Muller*, 43 F Supp 298; *Korzybski v Underwood & Underwood, Inc.*, 36 F2d 727 (2d Cir 1929)).

Some scholars regard this latter proposition to have been called into question by dicta in *Mazer v Stein*, 347 US 201, 217 (1954) ("Neither the Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted."), and by *In re Yardley*, 493 F2d 1389 (CCPA 1974) (design patent can issue on copyrighted work). See, for example, 1 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright*, § 2.19 (1991) ("*Nimmer on Copyright*"). Both of these decisions, however, involved design patent and copyright issues, an area in which Congress may have contemplated some degree of overlap. There is, however, no case holding that utility patents and copyrights can protect the same aspect of the same work. Several cases, *Baker v Selden* among them, express the contrary view. See also *Taylor Instrument Co. v Fawley-Brost Co.*, 139 F2d 98, 99 (7th Cir 1943). *Mazer* itself cites approvingly not only *Baker v Selden*, but two other cases in which the courts observed that plaintiffs should have sought a patent if they wanted to protect the creative aspect of their work which they were trying to protect through copyright. See note 71. The issue whether utility patents and copyrights can protect the same aspect of computer programs is currently hotly debated among intellectual property lawyers. See, for example, Perle, Meyer & Siber, 7 Computer L at 7-8 (cited in note 32); D.C. Toedt, Bonito Boats *Follow-up*, 6 Computer L 28 (July 1989).

This author contends that, notwithstanding the dicta quoted above, *Mazer v Stein* did not effect any significant change in the holding of *Baker v Selden*. The Court in *Mazer* merely decided that the copyright in a statuette (which qualified for protection as a "work of art") was not invalidated because of the subsequent reproduction of it to serve as a base for a lamp. The statuette served the same aesthetic function as a lamp base as it had as a free-standing sculpture. The Court in *Mazer* made clear that had the sculpture served a utilitarian function or had any utilitarian function been intermingled with its aesthetic function, it would have regarded the matter differently. 347 US at 212-13. The 1976 Act has codified this aspect of *Mazer*. See 17 USC § 101 (definitions of "pictorial, graphic, and sculptural work" and "useful article"). See also id at § 113. If anything, *Mazer* clarified that *Baker v Selden* should be understood as a case concerning the appropriate scope of protection for copyrighted works, an issue generally arising in the course of an infringement determination. See Reichman, 42 Vand L Rev at 693-95 n288 (cited in note 2).

[71] The unprotectability of abstract ideas is the proposition for which a number of courts have recently cited *Baker v Selden*. See, for example, *Cable/Home Communication Corp. v Network Productions, Inc.*, 902 F2d 829, 842 (11th Cir 1990); *Whelan*, 797 F2d at 1236; *Toro Co. v R & R Products Co.*, 787 F2d 1208, 1211-12 (8th Cir 1986); *Apple Computer, Inc. v Formula Intl, Inc.*, 725 F2d 521, 524 (9th Cir

infringe Selden's copyright when Baker wrote his own treatise, in his own words, describing the special double-entry method of bookkeeping."[72] This statement implies that the court in *Paperback* thought that Selden sued Baker for copyright infringement because of similarities in the explanatory material in Baker's and Selden's books. This was not so. Selden sued Baker for copyright infringement because Baker's book contained sample ledger sheets that were substantially similar in arrangement to those found in the Selden book.[73] *Baker v. Selden* was, in other words, a "nonliteral similarity" or "structure, sequence, and organization" case.[74] To put it in a slightly different way, Baker's book offered potential users of Selden's accounting system a substantially similar "user interface" to that of Selden's.

A review of the parties' arguments in *Baker v. Selden* reveals the parallels between the legal contentions in that case and those in *Paperback*. Selden, who had won in the lower courts, was, in effect, arguing to the Court that there was

1984); *Atari, Inc. v Midway Mfg. Co.*, 672 F2d 607, 615 (7th Cir 1982); *Rubin v Boston Magazine Co.*, 645 F2d 80, 82 (1st Cir 1981); *Reyher v Children's Television Workshop*, 533 F2d 87, 90 (2d Cir 1976).

Mazer v Stein, 347 US at 217, does seem to give *Baker v Selden* as an illustration of the copyright principle that copyright protection is available only for the expression of an idea, not the idea itself. Professor Nimmer relied on this aspect of *Mazer* to argue that *Baker v Selden* should be understood as limited to the idea-expression distinction. See 1 *Nimmer on Copyright* § 2.18[D] (cited in note 70). Professor Reichman has pointed out that the idea-expression distinction preexisted *Baker v Selden*, and if that was all the case represented, it would have been superfluous at the time it was rendered. Reichman has also observed that the important historical role of *Baker v Selden* "was to override the exclusive reproduction rights as applied to utilitarian works . . . [when] the standard defenses (including idea-expression) appeared insufficient to guarantee that a third party's right to use functional features embodied in the work . . . ." Reichman, 42 Vand L Rev at 693 n288 (cited in note 2).

That *Mazer* did not mean by this reference to *Baker v Selden* to effect a radical change in the meaning of that case is demonstrated by the fact that, in the same paragraph as its discussion of *Baker v Selden*, *Mazer* cited approvingly two other cases in which defendants had copied the functional designs depicted in plaintiffs' copyrighted drawings by constructing the functional work depicted in the drawing. The courts in these two cases applied the principles of *Baker v Selden* in ruling that the copyright had not been infringed. See *Muller*, 43 F Supp 298 (design for controlling the flow of automobile traffic approaching a bridge not protectable by copyright in drawings); *Fulmer v United States*, 103 F Supp 1021 (Ct Cl 1952) (copyright in drawings depicting design for parachutes not infringed by manufacture of parachutes embodying the design), cited in *Mazer*, 347 US at 217 n39. These cases indicate that by failing to seek patent protection for the functional designs depicted in the drawings, plaintiffs had lost the right to exercise control over the functional implementation of the design by others. *Muller*, 43 F Supp at 299-300; *Fulmer*, 103 F Supp at 1022. For further discussion of how *Baker v Selden* has been understood over time, see Reichman, 42 Vand L Rev at 693-695 n288 (cited in note 2).

[72] *Paperback*, 740 F Supp at 54. It is a small point, but neither Baker nor Selden had written a "treatise" on this accounting method. The Court described Selden's book as consisting of "an introductory essay explaining the system of bookkeeping referred to, to which are annexed certain forms or blanks, consisting of ruled lines and headings, illustrating the system and showing how it is to be used and carried out in practice." *Baker v Selden*, 101 US at 100.

[73] That Baker's ledger sheets were substantially similar to Selden's can be discerned from the Supreme Court's observation that if Selden's copyright extended to the accounting system, the Court would agree that Baker's book infringed the copyright. *Baker v. Selden*, 101 US at 100. The court stated that Baker's work "use[d] a similar plan so far as the results are concerned; but makes a different arrangement of the columns, and use[s] different headings." Id.

[74] Neither in *Paperback* nor in the *Whelan* decision, discussed in notes 30-52 and accompanying text, was there any recognition that *Baker v Selden* was a nonliteral similarity or SSO case in which the SSO was ruled outside the scope of copyright.

original expression[75] in the selection, ordering, and arrangement of the columns and headings of the ledger sheets contained in his copyrighted book.[76] That such elements could be protected by copyright law was evident, Selden argued, from cases involving maps, charts, and diagrams,[77] among others.[78] Selden insisted that Baker's arguments about "uncopyrightable subject matter" simply missed the point.

In support of his "uncopyrightable subject matter" defense, Baker pointed out that Selden had gone to the Patent Office to get patent protection for his bookkeeping system.[79] Baker asserted that this demonstrated that the *system* was not the proper subject matter of copyright, but should be protected, if at all, by a patent.[80] It was a contribution to the useful arts, Baker argued, not to literature. Baker insisted that the ledger sheets conveyed no thought, provided no information, and expressed no idea over and above the system they embodied.[81]

Baker relied on some cases denying copyright protection to forms,[82] as well as on the Court's then very recent decision in *The Trademark Cases*.[83] In that set of cases, the Court had very clearly distinguished between things that were "writings" of "authors," and hence within the subject matter of copyright, and those that were "inventions" in the "useful arts," which were the province of the patent system.[84] Baker argued that Selden, having been unable to get a patent, should not use copyright to get indirectly the kind of

[75] Selden's arguments are summarized in 25 LEd at 842, as well as described by the Court in the body of the opinion. The Court understood Selden to be contending "that the ruled lines and headings, given to illustrate the system, are a part of the book, and, as such, are secured by the copyright: and that no one can make or use similar ruled lines and headings, made and arranged on substantially the same system, without violating the copyright." 101 US at 101.

[76] See notes 139-142 and accompanying text for a discussion of selection, ordering and arrangement of categories of information in forms cases.

[77] See synopsis of Selden's argument, 25 LEd at 842.

[78] The Court's opinion indicates that Selden relied heavily on *Drury v Ewing*, a case in which copyright was claimed in a chart of patterns for clothing. *Baker v Selden*, 101 US at 107. The Court questioned the *Drury* decision, but concluded that in any event it was not a controlling case. Id.

[79] That Baker relied on this point is demonstrated in the summary of his argument in 25 LEd at 841. The Court's opinion indicated that no patent issued on Selden's system. 101 US at 104. The opinion does not reveal whether the patent was denied or merely withdrawn.

[80] More recent cases would suggest that Selden's system was not patentable subject matter because it was a business method, see, for example, *Ex Parte Murray*, 9 USPQ2d (BNA) 1819 (PTO Bd Ap 1988), although one case suggests the method might be patentable if carried out by computer. See *Paine, Webber, Jackson & Curtis, Inc. v Merrill Lynch, Pierce, Fenner, & Smith, Inc.*, 564 F Supp 1358 (D Del 1983).

[81] See synopsis of Baker's argument preceding the Court's opinion. 25 LEd at 841.

[82] Id at 842. See also 101 US at 106-07 for the court's discussion of a case that had denied copyright protection to a cricket scoring sheet.

[83] 100 US 82 (1879). In that opinion, the Court ruled that the constitutional clause empowering Congress to legislate to give exclusive rights to "authors" for their "writings" and to "inventors" for their "discoveries" in the "useful arts" did not give Congress power to pass a uniform national trademark statute. Baker's reliance on these cases is referred to at 25 LEd at 842.

[84] *The Trademark Cases*, 100 US at 94. Interestingly enough, the Court's opinion in *Baker v Selden* contains no direct reference to *The Trademark Cases*, although the Court's concern about not allowing copyright law to be used to protect things in the patent domain is evident from the *Baker v Selden* opinion, discussed in note 86 and accompanying text. It is worth noting that the Supreme Court's recent decision, *Feist Publications, Inc. v Rural Telephone Service Co.*, 111 S Ct 1282 (1991), relied heavily on and quoted approvingly from both *The Trademark Cases* and *Baker v Selden* concerning the meaning

protection that the Patent Office would not grant him directly. Even embodied in a copyrighted book, the system still remained outside the subject matter of copyright. As long as he wrote his own explanatory material about the system, Baker insisted that no copyright liability could arise from having substantially similar ledger sheets in his book, for he needed to be able to reproduce similar ledger sheets in order to illustrate the bookkeeping system in his own book.

The Supreme Court overturned the ruling in Selden's favor and ordered the complaint against Baker to be dismissed.[85] The Court agreed with Selden that he held a lawful copyright in the book he wrote explaining his accounting system.[86] But the Court agreed with Baker that the copyright in the book no more gave its owner exclusive rights in the accounting system than the copyright in a book on the composition and uses of medicines would give its author exclusive rights over manufacture and application of the medicines.[87] To get exclusive rights for innovations of these sorts, an innovator had to go to the Patent Office. "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."[88] From copyright, the Court insisted, an innovator could get only the exclusive rights to print and distribute his book about the art.[89]

of originality in expression and authorship in copyright law, as well as what copyright law can properly protect in a work.

[85] *Baker v Selden*, 101 US at 107.

[86] Id at 101-02.

[87] Id at 102. The Court noted that a book on an accounting system might "contain[] detailed explanations of the art, [and] it may be a very valuable acquisition to the practical knowledge of the community." Id. But this still did not make the detailed knowledge concerning the art a protectable element of the copyrighted book. Id. The last substantive line of the recent *Feist* decision quotes *Baker v Selden* on a similar point: " 'great praise may be due to the plaintiffs for their industry and enterprise in publishing this paper, yet the law does not contemplate their being rewarded in this way.' " *Feist*, 111 S Ct at 1297, quoting *Baker v Selden*, 101 US at 105.

[88] *Baker v Selden*, 101 US at 102. The Court pointed out that the novelty of the art being described in the book has no relevance in determining the scope of copyright protection in the book. Id. This aspect of the Court's opinion is worth noting because *Paperback* makes much of the "unobviousness" of the Lotus interface as if it had significant bearing on the protectability of it by copyright. The court even states that "obviousness" was one of five concepts to be considered in analyzing a case such as *Lotus v Paperback*, 740 F Supp at 58. (The others were the idea-expression distinction, functionality, originality, and merger. Id at 58-59.) "Obviousness" is an important concept in patent law, but not in copyright law. The court's frequent references to "obviousness" in *Paperback* indicate that the court may have been confused on this point. See, for example, *Paperback*, 740 F Supp at 58 (asserting that it would be wrong to deny copyright to the "most original and least obvious" aspects of a work). Id at 65 (asserting that although the core idea of a spreadsheet program was "functional and obvious," not "every possible method" of designing a spreadsheet program was obvious, and originality in a copyright sense involved "pressing beyond the obvious"). Id at 66 (use of "+" to represent addition was "*obvious* if not essential" (emphasis in the original)). Id at 68 (referring to the command structure of 1-2-3 as "original and nonobvious"). See also id at 79 (concerning the need to protect "strikingly innovative" aspects of programs by copyright). These statements suggest that the court in *Paperback* had lost sight of a fundamental point of the *Baker v Selden* opinion that the novelty (or nonobviousness) of Selden's accounting system had no bearing on whether it was protectable by copyright law.

[89] *Baker v Selden*, 101 US at 102-103. The Court explained:

As all this pertained to Selden's claims, the Court reasoned that since Selden's copyright did not give him any exclusive rights in the accounting system, Baker was' free to put similar ledger sheets illustrating use of the system in his book as well. To hold otherwise would indirectly give Selden exclusive rights in his system that the Court regarded as improper to recognize directly.[90] In effect, the Court decided that the selection, ordering, and arrangement of these columns and items in the ledger sheets were constituent parts of Selden's accounting system, and hence not a part of the book's protectable expression.[91]

> The copyright of a work on mathematical science cannot give to the author an exclusive right to the methods of operation which he propounds, or to the diagrams which he employs to explain them, *so as to prevent an engineer from using them whenever occasion requires.* The very object of publishing a book on science or 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* . . . .

Id at 104 (emphasis added) (Much of this passage was quoted with approval in *Feist*, 111 S Ct at 1290.). The Court indicated that the "useful art" was not protectable regardless of whether it was explained in a book or illustrated by diagrams: "Those illustrations are the mere language employed by the author to convey his ideas more clearly. Had he used words of description instead of diagrams (which merely stand in the place of words) *there could not be the slightest doubt that others, applying the art to practical use, might lawfully draw the lines and diagrams* which were in the author's mind, and which he thus described by words in his book." Id (emphasis added).

Cases such as *Whelan*, 797 F2d 1222, assert that they are consistent with *Baker v Selden* because if a particular detail of a program is necessary to achievement of a program's general purpose or function, idea/expression merger will be found. See note 40 and accompanying text. Indeed, the Third Circuit in *Whelan* purported to derive its test from *Baker v Selden*, id at 1235-36. *Whelan*, however, fundamentally misconstrues the larger meaning of *Baker v Selden*. The Whelan test would give "functional works" a far broader scope of copyright protection than would be available to artistic and fanciful works, a result out of keeping with traditional principles of copyright law. See notes 123-127 and accompanying text.

[90] *Baker v Selden*, 101 US at 104-05. The court noted that the plausibility of Selden's claim arose from the nature of the work the case involved: "In describing the art, the illustrations and diagrams employed happen to correspond more closely than usual with the actual work performed by the operator who uses the art." Id at 104. While in most instances useful arts were embodied in wood, metal, or stone, the peculiar art this case involved was embodied in a writing. But, the Court announced, "the principle is the same in all. The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself." Id at 105.

[91] This review of *Baker v Selden* helps to explain why the court in *Synercom*, 462 F Supp 1003, ruled that the manner in which Synercom had formatted data for input to its structural analysis program for engineering projects was a constituent part of the "idea" or "method" of the program. The *Synercom* opinion reflects that there were more than three million ways to order the data for input into the structural analysis program. Id at 1012. That did not, however, mean that there were three million ways to order the data that would be sound from an engineering standpoint for conducting structural analyses. If the data were ordered in accordance with a method Synercom had devised, the ordering of the data would very likely be inextricably interconnected with the method, and under *Baker v Selden*, would be part of the work's idea. This appears to be the basis of the court's ruling in *Synercom*.

Paperback, 740 F Supp at 55, characterizes the "central proposition" of *Synercom* as though nonliteral sequencing should always be treated as a circumstance in which idea and expression have merged. This is not the case, for, in a footnote, the court in *Synercom* indicated that in a proper case nonliteral elements of computer programs might be protectable by copyright. 462 F Supp at 1013 n5. *Synercom* did recognize, however, that in some cases the ordering of data might be reflective of a method, and when it was, the sequence was outside the scope of copyright. In the usual case, said the

Given that Baker may have made his sample ledger sheets somewhat different from Selden's out of fear that making them identical would almost inevitably bring on a lawsuit, it is worth inquiring whether there is anything in the Court's opinion suggesting that it would have ruled differently if Baker had copied Selden's ledger sheets exactly, column for column, heading for heading. The question has a clear bearing on the implications of *Baker v. Selden* for the *Paperback* dispute which involved at least some exact copying. The last substantive statement of the *Baker v. Selden* opinion directly answers this question: "[t]he conclusion to which we have come is, that blank account-books are not the subject of copyright; and that 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*."[92] The Court's decision, then, did not rest on a finding that Baker's ledger sheets were different enough from Selden's that they did not infringe. The Court ruled that Selden's ledger sheets were simply not protectable by copyright law, just as Baker asserted.

While a later section of this article will question whether there was sufficient expressiveness in the Lotus command structure to render it a protectable feature of the Lotus program,[93] the next subsection will explain why *Paperback* itself suggests that this aspect of the Lotus interface was a constituent part of a "system" that, under section 102(b), should have put it outside the scope of copyright law. The subsection will concentrate on the macro language issue because of the stronger indication in the opinion that this was a system whose elements had to be copied in order for others to use and "express" the same system. The court should also have inquired whether the Lotus command structure was a component of the larger Lotus system for managing spreadsheet functions, one that other spreadsheet program developers might have needed to copy in order to use the same system.[94]

court, "sequenc[ing], choice, and arrangement have only *stylistic significance*, rather than constituting as they would here, the essence of the expression." Id at 1014 (emphasis added).

[92] *Baker v Selden*, 101 US at 107 (emphasis added). It would have been easy for the Court to have disposed of the case by saying that Baker's ledger sheets were different enough to be noninfringing. The Court, however, decided to tackle the more difficult problem presented by the case about whether the ledger sheets as illustrative of the bookkeeping system were protectable by copyright. Its decision that the ledger sheets were part of the system, and hence unprotectable by copyright law, has deeply shaped American copyright law. See note 70.

One thing that is somewhat archaic about *Baker v Selden* is the emphasis both the parties and the Court gave to the issue of "copyrightability" of the subject matter at issue in the case. See notes 79-91 and accompanying text. See also notes 27-29 and accompanying text. At the time *Baker v Selden* was handed down, courts had not yet recognized news reports or advertisements as copyrightable subject matter. See 101 US at 105-06. It is partly because these classes of works were later ruled to be copyrightable subject matter that *Baker v Selden* is in need of an updated interpretation. See, for example, *Bleistein v Donaldson Lithography Co.*, 188 US 239, 251 (1903) (advertisement was copyrightable); *International News Service v Associated Press*, 248 US 215, 234 (1918) (reflecting view that AP newspapers could have been copyrighted).

[93] See notes 133-157 and accompanying text. See also note 150, which will suggest that irrespective of the macrocommand language issue, the Lotus commands may still have been constituent parts of the Lotus spreadsheet system.

[94] It is beyond the scope of this article to deal at length with Paperback's "standardization" defense. Other commentators have dealt with "standardization" defenses in computer program

B. The Macrocommand Language as a System

The original Lotus complaint asserted that the command and macro language and syntax of the 1-2-3 interface were protectable by copyright law.[95] Although the syntax claim seems not to have been directly pressed at trial,[96] it is evident from *Paperback* that Lotus was still asserting that the macrocommand language was protectable.[97] Paperback attempted to counter this claim by arguing that languages were unprotectable by copyright law and by offering evidence to show that the Lotus macro facility involved a language.[98] In view of the court's sharply negative reaction to the defense,[99]

copyright cases at greater length. See, for example, Karjala, 28 Jurimetrics J at 33 (cited in note 31); Menell, 41 Stan L Rev at 1045 (cited in note 6); Stern, 14 Colum-VLA J L & Arts at 283 (cited in note 13); Comment, 52 U Pitt L Rev at 689 (cited in note 13). Economists have argued that copyright law should be applied to computer program user interfaces so as to promote standardization. See, for example, Joseph Farrell, *Standardization and Intellectual Property*, 30 Jurimetrics J 35 (1989); Menell, 41 Stan L Rev at 1045 (cited in note 6).

The court in *Paperback* was surely wrong to say that there was no copyright precedent to support recognition of a standardization defense. 740 F Supp at 79. A number of computer program copyright cases have regarded elements of user interfaces as unprotectable for reasons consistent with a standardization defense: some because the elements had become conventional to programs of the sort the case involved, see, for example, *Telemarketing Resources v Symantec Corp.*, 12 USPQ2d (BNA) 1991 (ND Cal 1989); some out of concern for users who otherwise would have to be retrained, see, for example, *Synercom*, 462 F Supp 1003; and some because market factors required commonality in user interfaces, see, for example, *Plains Cotton*, 807 F2d 1256. See also *Kepner-Tregoe*, 203 USPQ (BNA) at 132 (discussing how market factors can narrow the range of available expressions). In the context of traditional kinds of literary works (novels and plays), the *scènes à faire* doctrine is a comparable defense. See *Paperback*, 740 F Supp at 59 (recognizing the *scènes à faire* doctrine).

What may have made Paperback's argument so difficult for the court to accept was that it was an argument that virtually all of the Lotus interface had become a standard. Intellectual property scholars who reached consensus on many user interface issues at the LaST Frontier Conference were unable to reach consensus on what if any weight should be given to standardization concerns in analyzing copyright infringement claims involving user interfaces. *LaST Frontier Report*, 30 Jurimetrics J at 28, 31 (cited in note 6).

[95] See Complaint of Lotus Development Corp., discussed in note 18 and accompanying text. That the "language" defense was one of Paperback's principal defenses is evidenced by the extensive treatment given to the issue in its pretrial brief. See Pretrial Brief of Defendants Paperback Software Intl and Stephenson Software, Ltd., *Lotus Development Corp. v Paperback Software Intl*, CA No 87-0076-K 30-43 (Feb 1, 1990).

In the course of the litigation, Lotus seems to have downplayed the language claims, focusing more attention on the "command structure" issue for which it may have been easier to find helpful copyright precedents and on the protectability of the user interface as a whole.

[96] Syntax is, by definition, a set of abstract rules which must be followed for statements in a language to be meaningful. Since it is well established that "rules" are not protectable by copyright law (see, for example, *Morrissey v Proctor & Gamble*, 379 F2d 675), a direct claim for the Lotus syntax as a copyrightable element of the program would be on shaky grounds. Lotus syntax, because of its import in the ordering of commands in the command structure, remained indirectly in the case. For the court's recognition of this, see note 101 and accompanying text. That syntax affects semantics can be easily demonstrated with an example in the English language. "The dog bit the man" has a different meaning than "the man bit the dog," notwithstanding the fact that both sentences have the same words. The meanings differ because of English syntactic rules about the placement of subjects of sentences vis-à-vis verbs and other parts of the predicate.

[97] *Paperback*, 740 F Supp at 63.

[98] Professor Harry W. Lewis of the Harvard University Computer Science Department submitted an affidavit directed to this issue to support Paperback's defense.

[99] The court's lack of regard for this argument is best revealed by the fact that the "language" defense is only discussed at the end of the opinion in a section of the opinion entitled "A Postscript

it would seem that the defense may not have been focused enough to aid the court in appreciating its copyright significance.

Had the court understood that the language defense was, in fact, a section 102(b) "system" defense, it might not only have integrated the analysis of this issue into the body of an infringement analysis, but might even have been able to perceive the connection between this defense and the command structure issue that was so central to the ruling in *Paperback*.

Paperback contains two paragraphs describing the macro facility of Lotus 1-2-3. This discussion reveals that the court understood how users of the Lotus program could use the Lotus commands to construct macros in order to adapt the program to serve their needs better:

> Rather than going step-by-step through the same sequence of commands each time there is a need to perform a particular function, the user may store a sequence of command terms as a "macroinstruction," commonly called a "macro," and then, with one command stroke that invokes the macro, cause the programmed computer to execute the entire sequence of commands.[100]

The Lotus commands could, in other words, be used as building blocks for construction of these macros, which is why they could accurately be called elements of a language. After noting that macros could be built not only by combining Lotus command terms into sequences, but also by combining function keys and other aspects of the interface into the sequences, the court goes on to make this very revealing statement: "[b]ecause macros may contain many menu choices, *the exact hierarchy*—or structure, sequence, and organization—*of the menu system is a fundamental part of the functionality of the macros*."[101]

This statement demonstrates that the court recognized that the structure of the commands was part of the Lotus macro system, which, if one was taking section 102(b) seriously, would need to be regarded as outside the bounds of

on the Nature of Decisionmaking in This Case" under a subsection entitled "Strained Analogies and Word Games." *Paperback*, 740 F Supp at 71.

As with Paperback's functionality defense, the court was critical of Paperback's attorneys for failing to make a coherent statement of the language defense, see note 163, and ultimately made its own statement of the argument underlying this defense. *Paperback*, 740 F Supp at 72. The court offered several reasons for regarding the argument as flawed, id at 72-73, among them, that the defendants "though invited at trial to do so, have cited no precedent that supports [this] contention . . . ." Id. But see note 107. The court thought that the argument depended on "arbitrary definitions of words, adopted for undisclosed reasons," and concluded that this defense was "totally without merit." Id.

Earlier sections of *Paperback* mention machine, assembly, and higher order computer languages. Id at 44-45. There is no mention of these kinds of languages in the subsection discussing Paperback's functionality defense. The court may, however, have regarded the Lotus commands as too different from these other kinds of languages to be a comparable phenomenon. If, as a matter of computer science, the Lotus macro facility contained elements satisfying the formal definition of "language," the court should have taken the defense seriously.

[100] Id at 64.

[101] Id at 65 (emphasis added). See note 96 concerning the embedding of syntax in the Lotus command structure. See also *Paperback*, 740 F Supp at 78 (referring to Lotus's "command facility" and "macrocommand facility," and the availability of "translation devices" to allow 1-2-3 macros to be converted in other programs). The statement quoted in the text was made in a section of the opinion describing the Lotus interface, id at 63-65, and before any discussion of what the court considered to be copyrightable expression in the interface.

copyright protection.[102] No further reference to this issue can be found in the court's analysis of the "copyrightability" of the Lotus 1-2-3 interface or of the lawfulness of Paperback's copying of elements of the interface. Nor is there a reference to this issue in the part of the opinion discussing the language defense.[103]

Even apart from this statement, Paperback's language defense should have been taken more seriously. By definition, a language is a formal system consisting of three elements: a vocabulary, a syntax, and semantics.[104] There was evidence in the record demonstrating that, as a matter of computer science, the Lotus commands could properly be understood to be elements of a language.[105] *Paperback* gives no reasons for its rejection of this evidence.

The court's treatment of the question of whether copyright protection is available for a language leaves much to be desired. The court challenged Paperback to find precedents to show that languages were uncopyrightable.[106] From Paperback's failure to find any, the court seems to have inferred that the defense had no merit. Instead, the court might have insisted that Lotus affirmatively prove that languages are copyrightable, or, in the alternative, it might have recognized that Lotus was bringing forth a novel question of law for it to decide. Either choice likely would have led to a clearer understanding of the copyright issues raised by this assertion.

There is some copyright precedent that might support the view that languages are, in fact, not protectable by copyright law even when embodied in the texts of copyrighted works. The "shorthand system" cases, like the *Paperback* case, involved claims to language components used in the practice of the plaintiffs' systems.[107] The courts in these cases have, with the obligatory nods to *Baker v. Selden*, denied plaintiffs the protection they sought from copyright law. None of these cases is discussed in the subsection on the language defense.[108] Nor does the court mention the views of a number of

[102] There was no reference to section 102(b) in the part of the opinion that discusses the language defense. Properly understood, Paperback's "compatibility" argument should be seen as a component of the "language as system" defense. See notes 110-112 and accompanying text for a discussion of compatibility issues.

[103] *Paperback*, 740 F Supp at 73-75.

[104] See Inst of Electrical and Electronics Engineers ("IEEE"), *Standard Dictionary of Electrical and Electronic Terms* 478 (IEEE, 3d ed 1984). See generally Richard H. Stern, *On Copyright in Computer Languages*, 17 Rutgers Computer & Tech L J 321 (1991).

[105] See note 98. Moreover, Lotus itself had referred to 1-2-3's command menu and macro language as protectable elements of the Lotus interface. See note 18 and accompanying text.

[106] 740 F Supp at 72.

[107] See, for example, *Brief English Systems*, 48 F2d 555 (finding no copyright infringement where similarities between the plaintiff's and defendant's works were attributable to their being about the same shorthand system, citing *Baker v Selden*, 101 US 99); *Griggs v Perrin*, 49 F 15 (same). See also *Signo Trading Intl v Gordon*, 535 F Supp 362 (translation system not protectable by copyright). But see *Nikanov v Simon & Schuster, Inc.*, 246 F2d 501 (2d Cir 1957) (finding infringement of a Russian alphabet and language guide).

[108] This part of *Paperback* makes a passing reference to the "coined words" of *Reiss v National Quotation Bureau, Inc.*, 276 F 717 (SDNY 1921), but does not discuss the case. *Reiss* was an uncharacteristically terse opinion written by then District Judge Learned Hand in which the court upheld the validity of a copyright in a compilation of coined words intended to enable purchasers of the book to encode their communications to other persons via cable. It was, in short, a book

commentators who have opined that computer languages should be regarded as unprotectable by copyright law.[109]

Analysis of Paperback's "compatibility" defense should also have been integrated with the "language-as-system" defense.[110] The macro facility of the Lotus program allows users with complex computational needs to create "libraries" of macros that allow rapid single stroke execution of the sequences necessary to accomplish these tasks. Whole books have been written to advise

consisting of a possible vocabulary without syntax or semantics. *Brief English Systems v. Owen*, 48 F2d 555 (2d Cir 1931), decided ten years after *Reiss* and by the Court of Appeals for the Second Circuit, seems a more apposite precedent, given its more linguistic character.

Compilations have long been troublesome for copyright law, and the temptation to protect them on account of the work that was required to prepare them has proved very strong over the years, as demonstrated by *Reiss* and the long line of "sweat of the brow" cases recently rejected as erroneous interpretations of copyright law in *Feist*, 111 S Ct 1282, 1292-95. Still, copyright does have a long history of protecting compilations. See generally Jane C. Ginsburg, *Creation and Commercial Value: Copyright Protection of Works of Information*, 90 Colum L Rev 1865 (1990).

[109] For commentators expressing this view before *Paperback* issued, see, for example, John P. Sumner, *The Copyright/Patent Interface: Patent Protection for the Structure of Code*, 30 Jurimetrics J 107, 112 (1989); Leo J. Raskind, *The Uncertain Case For Special Legislation Protecting Computer Software*, 47 U Pitt L Rev 1131, 1174 (1986); Richard H. Stern, *The Bundle of Rights Suited To New Technology*, 47 U Pitt L Rev 1229, 1239 n64 (1986); Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 Mich L Rev 866 (1990) (authored by Steven R. England). This Note was cited in *Paperback*, 740 F Supp at 53. It asserts that the *Synercom* decision was correct because the input formats for Synercom's format were a language: "Thought of in this way, *Synercom* was an easy case, for there can be no more protection for input formats than for the English language itself." Id at 882 n82. See also Stern, 14 Colum-VLA J L & Arts at 283 (cited in note 13); Comment, 39 Emory L J at 1293 (cited in note 13).

[110] Discussion of Paperback's compatibility defense can be found at *Paperback*, 740 F Supp at 69, 77-79. Paperback argued that in order to develop a commercially viable spreadsheet program, it was "necessary" for its program to be "compatible" with Lotus. Copying the Lotus command hierarchy was said to be necessary to achieve compatibility. The court found it easy to dispense with this argument by pointing out that there were other commercially viable spreadsheet programs that had different command structures than Lotus, and hence Paperback had not proven the necessity of copying. Id at 69. (It is worth noting that the court had found "necessity" as to some features of the Lotus interface, such as the use of the "/" to invoke the command menus and the rotated "L" grid, despite the fact that there were programs on the market with different ways of doing these functions. See notes 49-50 and accompanying text.)

Yet the opinion goes on to suggest that if Paperback had copied the Lotus commands in a more indirect or inefficient manner—as by using a help facility to inform users what the equivalent Paperback command was for each Lotus command, or having a macrocommand conversion facility like that used by Excel—it might have achieved compatibility in an acceptable manner. Id. Use of one of these alternative ways to achieve compatibility would, of course, have also involved a significant degree of copying of the command structure of Lotus's interface. Yet, the court suggests—without saying so directly—that it would have ruled this more indirect and inefficient copying to be legal. Although the court identifies these two seemingly noninfringing ways to achieve compatibility, the court's "necessity" test, as stated earlier in the opinion, seemed to recognize that the existence of only a small number of alternative ways to do something would satisfy the merger test. See id at 66. Merely two alternatives to Paperback's copying of the command structure does not seem to be more than a small number of alternatives.

From a user's standpoint, the inefficiency of having to call on a help screen to identify which VP-Planner commands were equivalent to which Lotus commands seems an unwarranted nuisance, requiring difference for difference's sake, rather than because of some expressive quality in the command terms. See *Kepner-Tregoe*, 203 USPQ (BNA) 124. Recall that the Supreme Court in *Baker v Selden* indicated it would have ruled no differently in that case had Baker copied Selden's ledger sheets exactly. See note 90 and accompanying text. A recent case ruling that competitors were not required to engage in inefficient copying when traditional principles of federal intellectual property law would regard the copied aspect of the work to be unprotectable was *Bonito Boats, Inc. v Thunder Craft Boats, Inc.*, 489 US 141 (1989).

users on how to construct macros for specific kinds of functions for Lotus 1-2-3.[111] After investing considerable time and energy in the creation of macros, users will naturally want to continue to be able to use these macros, and perhaps to share them with others with whom they may be working and who may or may not have a Lotus program on their computers.

"Compatibility"[112] with the Lotus program, in essence, would allow users with macro libraries to continue to enjoy the fruits of their own labor by allowing them to "port" over to another spreadsheet program the macros they have constructed. As the court's description of the Lotus macro facility reveals, the exact same hierarchy of commands as 1-2-3 *must* exist in another program for macros built in the 1-2-3 system to function at all. While the macro facility of 1-2-3 is a highly useful aspect of the Lotus program—one that might well be patentable under today's standards for computer program-related inventions[113]—it is, at base, a system, and, for this reason, should be deemed to be outside the scope of copyright protection. To the extent the command structure is an essential component of the macro facility, it too may be outside the scope of copyright protection.

Only if the court determined that there was "expression" in the Lotus interface, over and above the role of significant components of it in the macro system, could it, consistently with the traditional principles of copyright law,

[111] See, for example, David P. Ewing, *1-2-3 Macro Library* (Que Corp, 3d ed 1990).

[112] "Compatibility" has been heatedly debated as a computer program copyright issue both in the United States and abroad. The issue arises both as to "internal interfaces" of computer programs and as to "user interfaces" of programs. From the standpoint of the technical community, the compatibility issues raised by both are much the same, even though copyright lawyers tend to treat them as somewhat more distinguishable. See, for example, *LaST Frontier Report*, 30 Jurimetrics J at 21-22, 26-31 (cited in note 6).

Some think that copyright law should be interpreted so that "interfaces" would be "ideas" and only the code implementing them should be considered copyrightable "expression." Under this view, if one firm copies the interface of another firm (at least nonfanciful aspects of it) in order to make a compatible product, no copyright infringement should be found. See, for example, Karjala, 28 Jurimetrics J 33 (cited in note 31); Michael A. Jacobs, *Copyright and Compatibility*, 30 Jurimetrics J 91 (1989). Others regard interfaces as valuable "nonliteral" elements of copyrighted programs, the copying of which should be treated as an appropriation of "expression." There is, in this view, no "right" under copyright law to make a "compatible" product. See, for example, William T. Lake, John H. Harwood & Thomas P. Olson, *Tampering With Fundamentals: A Critique of Proposed Changes in EC Software Protection*, 6 Computer L 1 (Dec 1989).

The recently adopted European Directive on copyright protection for computer programs recognizes that interfaces may be unprotectable "ideas" of computer programs, and allows such interfaces as are necessary to achieve interoperability to be copied without copyright liability. See *Council of the European Communities Directive on the Protection of Computer Programs*, reproduced in 42 Patent, Trademark & Copyright J (BNA) 109 (May 23, 1991). The case law in the United States is somewhat more mixed on this question. See cases discussed in the articles cited earlier in this note. The court in *Paperback* must have been aware of the larger debate over the "compatibility" issue, yet the opinion does not refer to this debate or to the case law that addresses the issue.

[113] Although the case law status on the patentability of computer program-related inventions is somewhat unclear, the U.S. Patent and Trademark Office has been taking a broad view of the patentability of such inventions. The Lotus macro facility would seem to qualify for patent protection under recently issued guidelines. See *PTO Report on Patentable Subject Matter: Mathematical Algorithms and Computer Programs*, 38 Patent, Trademark, & Copyright J (BNA) 563, 569-71 (1989). Concerning the case law on this issue, compare Donald S. Chisum, *The Patentability of Algorithms*, 47 U Pitt L Rev 959 (1986); Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 Emory L J 1025 (1990).

find infringement in *Paperback*. It is to the question of whether there was anything expressive in the Lotus interface that we now turn.

V
WHAT WAS "EXPRESSIVE" ABOUT THE LOTUS INTERFACE?

Determining what in a copyrighted work is "idea" and what is "expression" can be among the most difficult conceptual tasks faced by federal judges. For many kinds of works, such as novels and dramatic plays, there are numerous precedents through which a judge may search to find comparable situations to serve as a basis for making a judgment on the matter before him or her. In some cases, however, there is very little apart from general principles of copyright law to give a court guidance about how a particular case should be resolved. Judge Keeton perceived *Lotus v. Paperback* to be such a case. The court found nothing in the statute,[114] the legislative history, or the CONTU report that gave more than indirect guidance on the issues.[115] In the court's view, there were no prior cases that presented anything more than general similarities to the idea/expression problem the court found in *Paperback*.[116]

[114] Notwithstanding the court's statements early on in the opinion about the starting place of any analysis being the statutory language itself, the court only mentions the statutory definition of "computer program" in a background section of the opinion, *Paperback*, 740 F Supp at 50, and pays very little attention to § 102(b) (see notes 59-61 and accompanying text). The statutory definition of "computer program" more clearly supports Paperback's position in the litigation than Lotus's (see note 31 and accompanying text) as does the text of § 102(b) and the case law underlying it.

[115] *Paperback*, 740 F Supp at 50 (CONTU did not directly address the issue presented in *Paperback*). The opinion indicates that the court was aware of the differences of opinion among CONTU commissioners and staff about CONTU's views as to whether nonliteral elements of programs would be protected by copyright law. Id at 50-51. The opinion does not contain any reference to the legislative history of the 1980 software amendments. The court does refer in the legal background section to some statements from the legislative history of the 1976 Act concerning section 102(b), id at 49; but as noted above in notes 59-61 and accompanying text, the court pays very little attention to them, even though the text of § 102(b) and the legislative history concerning it were directly relevant to the controversy.

There is one place in *Paperback*, apart from the early background sections, in which the court makes use of legislative history to respond to a Paperback defense. In rejecting one of Paperback's "policy" arguments, the court referred to the testimony of a witness at a legislative hearing during the copyright revision process leading up to the 1976 Act. This witness had warned that copyright protection for computer programs might have disastrous consequences for standardization in that field. Id at 76. Because Congress decided to protect computer programs despite such warnings, the court regarded Paperback's standardization argument to be contrary to congressional intent. This statement, however, was made at the same hearing at which another witness, Professor Miller, warned of dire consequences from protecting computer programs by copyright law unless Congress adopted what became § 102(b) of the 1976 Act. See note 4. Given that Congress adopted § 102(b) in response to the concerns stated at this hearing, it may be that Congress thought § 102(b) could be used to address standardization concerns as well.

[116] Although *Paperback* contains some sections in which prior computer program copyright cases are either briefly summarized or quoted parenthetically (see, for example, id at 55), there is remarkably little analysis of prior cases in the substantive sections of the opinion analyzing the "copyrightability" of the Lotus interface and Paperback's copying of what the court found to be copyrightable elements of the Lotus interface. Occasional references are made to other cases, but only at a very general level. See, for example, id at 65 (brief reference to *Nichols*), id at 66 (citation to *Morrissey*), id at 67 (brief reference to *Whelan*), and id at 68 (brief reference to *Midway/Bandai*). The only prior computer program case that is discussed in some depth is *Softklone*, 659 F Supp 449, whose

The court regarded itself as left only with general principles of copyright law and overall congressional objectives as to the protection of computer programs with which to resolve the dispute.[117] In Judge Learned Hand's "patterns of abstractions" test, the court said it found a framework for sorting out idea from expression.[118] But the court got precious little help from the lawyers in the case in conceptualizing what an abstractions scale for a spreadsheet program might look like, and where on such a scale the Lotus interface (and its important component, the command structure) was to be found. Indeed, the court expressed frustration with the lawyers for taking "extreme positions"[119] that gave it no aid in constructing the abstractions scale. Lotus's lawyers apparently argued that every detail of its user interface was "expressive" (because of different ways functions could be done), and Paperback's lawyers argued that all aspects of the user interface were "ideas."[120] So, instead of building a fairly elaborate scale of abstractions, as Hand's test suggested was proper,[121] the court ended up with a dichotomy in which the only perceived generality at the "idea" pole was that of the electronic spreadsheet, and all of the particularities of the Lotus interface were lumped together at the "expression" pole.[122]

One of the traditional principles of copyright law that the court did not, but should have, employed in constructing its framework for analysis of the copyright issues was one that recognizes that the scope of a copyright (that is, the breadth of protection it provides and how far down the scale of abstractions it is appropriate to draw the line between idea and expression) tends to vary according to the nature of the work under consideration. Over the years, courts have come to perceive that there are differing levels of "expressive" content in different kinds of works. Highly fanciful or artistic

holding that computer program menu screens were separate works from the program was rejected in a section of the opinion discussing Paperback's subject matter jurisdiction defense. *Paperback*, 740 F Supp at 80-81. Even the *Whelan* decision, whose test for infringement heavily influenced the court's analysis of *Paperback* (see notes 36-52 and accompanying text), is not discussed much in the opinion.

[117] The court discusses what it considered to be relevant general principles of copyright law at 740 F Supp at 51-52 and 58-62 and relevant congressional objectives and policies at 52-53 and 73-79.

[118] *Nichols*, 45 F2d at 121, quoted in *Paperback*, 740 F Supp at 60.

[119] The court felt strongly enough about how little help it got from the lawyers that it devoted a subsection of the opinion to the issue. Id at 62. The court complained that by taking extreme positions, the lawyers had failed to aid the court in building an abstractions scale and assessing where along that scale various aspects of the Lotus interface were to be found. It warned that the "argument of an advocate who presses too far in one or the other of these [extreme] directions . . . will not only lose the argument advanced but also lose credibility for later advancing a more sensible alternative that proposes a less extreme but still favorable position along the scale." Id. Although the lawyers for both parties were criticized in this subsection for taking extreme positions, far more criticism in the opinion is directed at the defense lawyers. Some of Paperback's most vigorously asserted defenses were dismissed by the court as "word games." Id at 71-73, 79.

[120] Id at 62.

[121] See id at 60 (" 'Upon any work, and especially upon a play, a *great number of patterns* of increasing generality will fit equally well, as more and more of the incident is left out,' " quoting *Nichols*, 45 F2d at 121 (emphasis added)).

[122] The court found some aspects of the Lotus interface were "necessary" components of such an interface, and were eventually ruled to be "ideas" as well. See notes 49-50 and accompanying text.

works are generally regarded as enjoying a broad scope of copyright protection because of their predominantly expressive character.[123] Factual works generally have a narrower scope of protection; in order to further copyright's constitutional purpose of promoting the growth and dissemination of knowledge, the facts, theories, and other discoveries that such works contain are considered outside the scope of copyright.[124] Because of this, there is generally less expressive content to be found in factual works than in artistic or fanciful works. An even narrower scope of copyright protection is available for functional writings, such as rulebooks, forms, manuals for operating power plants, engineering drawings, and the like.[125] In general, only exact or near-exact copying of such works will be infringing, for the contents of these kinds of works tend to be predominantly functional.[126] Under the principles of *Baker v. Selden* and its progeny, this functional content is outside the scope of copyright protection.[127] Only if, and to the extent that such works contain some "expressive" content, is copyright protection available to their authors.

Although user interfaces of computer programs can be highly fanciful or artistic in character—videogames being perhaps the clearest example—many are more functional in character. Some may be too functional to be protectable by copyright.[128] That this should be so is not surprising in view of the fact that computer programs themselves are properly regarded as functional writings,[129] and the role of user interfaces is to provide users with access to program functionalities.[130] The aspects of user interfaces that are most readily accommodated by copyright are those that display words and symbols on computer screens. This explains why most of the user interface copyright cases have focused their copyright analysis on "screen displays," that is, on similarities in wording or graphical elements on computer

[123] Novels, dramatic plays, and cartoons are examples of works generally enjoying such a broad scope of protection. See, for example, *LaST Frontier Report*, 30 Jurimetrics J at 18 (cited in note 6).

[124] Biographies, histories, scientific reports, and fact compilations are among the works that enjoy this "thinner" protection under copyright law. Id. See also *Feist*, 111 S Ct 1282, 1289 (indicating that copyright protection for fact compilations is "thin").

[125] Elsewhere the author has distinguished between the truly functional character of computer programs and the kinds of "functional writings" mentioned in the text. Samuelson, 1984 Duke L J at 727-49 (cited in note 5). Manuals for operating power plants explain how the plant should be operated; they do not in themselves operate the plant. The wording used to explain the plant operations is the copyrightable "expression" in the manual. The details of plant operations described in the manual are among the work's "ideas." The only "function" of the manual is to convey information, a kind of function that does not disqualify a work from copyright protection. See notes 164-170 and accompanying text. This is in contrast to a computer program written to control the plant's operations which actually controls the functioning of the plant. The inherently functional nature of programs is what make them so different from traditional categories of copyrightable works.

[126] *LaST Frontier Report*, 30 Jurimetrics J at 18-19 (cited in note 6). See also cases cited in note 70.

[127] See note 70 and accompanying text.

[128] One example of this might be the digital display panels of gas pumps. See *LaST Frontier Report*, 30 Jurimetrics J at 27 (cited in note 6).

[129] Id at 19.

[130] See, for example, Bill Curtis, *Engineering Computer "Look and Feel": User Interface Technology and Human Factors Engineering*, 30 Jurimetrics J 51 (1989).

screens.[131] More abstract elements of user interfaces, such as the pairing of particular functions to particular keys on the computer keyboard, have generally been viewed as outside the copyright realm.[132] By abjuring the prior case law's focus on "screen displays" and embracing the copyrightability of "user interfaces," the court in *Paperback* may have opened the door to a considerable broadening of the scope of copyright protection for user interfaces.[133]

[131] See, for example, *Whelan*, 797 F2d 1222; *Manufacturers Technologies*, 706 F Supp 984; *Broderbund*, 648 F Supp 1127.

[132] See, for example, *Manufacturers Technologies*, 706 F Supp 984 (infringement based on similarities in wording and placement of words on screens; similarities in navigational elements of the user interface ruled not protectable); *Telemarketing Resources*, 12 USPQ2d (BNA) 1991 (rejecting claims involving certain pairings of keys to functions).

[133] *Paperback*, 740 F Supp at 79-80 (infringement said not to be based on screen display similarities, but upon similarities in user interfaces). The court quotes Lotus's definition of the Lotus user interface as including "such elements as 'the menus (and their structure and organization), the long prompts, the screens on which they appear, *the function key assignments, [and] the macro commands and language*.' " Id at 63 (emphasis added). As shown above in notes 49-50 and accompanying text, the court found some aspects of the Lotus interface to be unprotectable by copyright.

Few issues of computer program copyright law have confused the courts (and the Copyright Office) more than attempting to grasp the relationship between a computer program and a user interface. To enable the reader to understand the potential significance of the *Paperback* ruling concerning copyright protection for "user interfaces," it may be helpful to review how the issue had previously been dealt with by the courts and the Copyright Office.

The story starts with the Copyright Office decision in the early 1980s to begin registering videogame programs as audiovisual works. See, for example, *Stern Electronics, Inc. v Kaufman*, 669 F2d 852 (2d Cir 1982). Later, the Office began to issue separate registration certificates for the videogame programs. See, for example, *Williams Electronics, Inc. v Artic Intl, Inc.*, 685 F2d 870 (3d Cir 1982). Videogame case law distinguished between program and audiovisual copyrights; for in some videogame cases, a defendant would be found to have infringed the audiovisual, but not the program copyright; in other cases a defendant would be found to have infringed the program copyright, but not the audiovisual copyright. Sometimes both kinds of infringements were found. See, for example, *Williams Electronics*, 685 F2d 870; *M. Kramer Mfg. Co. v Andrews*, 783 F2d 421 (4th Cir 1986); *Midway Mfg. Co. v Strohon*, 564 F Supp 741 (ND Ill 1983). Registration practices and case law of this sort seemed to suggest that a firm needed one copyright to cover the program, and a second for whatever aspects of the program might be displayed on the screen.

Whelan was the first nonvideogame computer program copyright case in which user interface similarities became an issue. The trial court in *Whelan* relied heavily on similarities between the screen displays produced by the plaintiff's and defendant's programs as a basis for concluding that Whelan's copyright had been infringed. See *Whelan*, 609 F Supp at 1322. On appeal, Jaslow pointed out that Whelan was not charging infringement based on screen display similarities. Rather, her claim was that Jaslow had copied aspects of the underlying program. Jaslow noted that very differently structured programs could produce substantially similar or identical screen displays, and hence the trial judge had erred in relying on screen display similarities as the basis for finding copyright infringement as to the underlying program. See *Whelan*, 797 F2d at 1242-45.

The Third Circuit agreed with Jaslow's argument, but only up to a point. The court agreed that relying solely on screen display similarities to prove infringement as to structural elements of the underlying program would be error, but decided it was not error to consider screen similarities as some evidence of copying of underlying program elements. Because there was other evidence in the record that the court regarded as supporting Whelan's claim of copying of underlying structural elements of the program, the appellate court affirmed the infringement ruling.

Broderbund, 648 F Supp 1127, was the next case in which user interface/screen display similarities arose in a nonvideogame software copyright case. Here the infringement claim was based exclusively on screen display similarities, specifically the choice and arrangement of command terms on a series of menu screens for a printing program. Although the court was somewhat equivocal about whether Broderbund had an audiovisual or a program copyright or both, it found comfort in the *Whelan* decision, which it interpreted as a screen display similarity case. The court in *Broderbund* used the

A. The Proper Kind of Expressiveness Inquiry

Notwithstanding *Paperback's* lengthy abstract discussion of the copyright term "expression," there is virtually no discussion in it about what the court found to be "expressive" in an important element of the Lotus interface, the

Whelan test for software copyright infringement. This allowed the court to sidestep the issue of what kind of copyright it was dealing with.

Softklone, 659 F Supp 449, like *Broderbund*, involved claims of infringement based solely on user interface/screen display similarities, that is, the arrangement of command terms on menu screens. The judge in *Softklone* rightly pointed out that the court in *Broderbund* had misinterpreted *Whelan* by characterizing it as a screen display similarity case. The *Softklone* court took more seriously the points that different programs could produce the same screen displays and that substantially similar programs could produce different screen displays than had the appellate court in *Whelan*. The court in *Softklone* ruled that the program and the user interface were separate works requiring separate copyrights, a result consistent with the videogame cases. Fortunately for the plaintiff, it had obtained a separate copyright for its main menu screen as a compilation of terms. The court found infringement because the defendant had arranged the command terms in the same way as the plaintiff on one portion of the menu screen, and because the defendant had also used the same highlighting and capitalization of the first two letters of the command terms as had the plaintiff. (Interestingly, the court ruled that it was not an infringement to have the same set of command terms as another program, but only to arrange them in the same way when they could be arranged differently.)

Not surprisingly, the *Softklone* decision, insofar as it ruled that separate copyrights were necessary to cover user interface elements of programs, made some software developers quite nervous. Soon thereafter, the Copyright Office initiated an inquiry about the separate registration issue. See Notice of Inquiry, 52 Fed Reg 28, 311 (1987). Although the Office held hearings at which witnesses discussed reasons that separate registrations of programs and screen displays might be desirable, the Office decided against separate registrations. This may have been more as a matter of administrative convenience (thereby minimizing the number of forms the Office had to process) than because the Office had definitively resolved the copyright dilemma about the proper characterization of the relationship between computer programs and user interfaces. See Hearings Before U.S. Copyright Office on Registration and Deposit of Computer Screen Displays (Sept 9-10, 1987). Current Copyright Office policy is that if the predominant character of a program is audiovisual (such as a videogame), the program should be registered as an audiovisual work. If not, it should be registered as a literary work. Regardless of which registration category is used, the Office considers the program and its screen displays to be one work. The Office has left to the courts the task of deciding what elements of the program or the user interface are protectable by copyright law. Notice of Registration Decision: Registration and Deposit of Computer Screen Displays, 53 Fed Reg 21,817 (1988).

This, then, was the state of the law on this issue when the *Paperback* case was decided. Paperback, not surprisingly, relied on the *Softklone* decision to support its contention that the Lotus interface was unprotected by the copyright Lotus had registered for the program. Lotus, of course, thought that the Copyright Office's "one work" policy statement reflected the proper legal approach. Although some aspects of the prior user interface case law supported Lotus's position, none of the prior cases seemed to have been completely suitable to its purposes. Lotus seems to have decided to take a bolder and more direct approach to litigating its copyright infringement claim against Paperback, seeking protection not just for the screen displays 1-2-3 generated (the issue as to which the separate registration controversy had arisen), but rather for its "user interface" as an important part of the program.

The court was persuaded by Lotus that the user interface of Lotus 1-2-3 was a copyrightable element of the program, the copying of which was unlawful. *Paperback* makes a point of distinguishing its ruling in this respect from the prior user interface cases, particularly *Softklone*, seeming to regard what it had done as breaking new ground for copyright law. This decision to conceptualize the user interface as a protectable element of a program may help to explain why there is so little discussion of or reference to the other user interface/screen display cases in *Paperback*.

Had the court not found the Lotus interface to be a copyrightable element of the Lotus program, it is worth noting that Paperback might not have won the case as a whole. The opinion indicates that the second phase of the trial would have determined whether Paperback copied copyrightable elements from the source or object code of the Lotus program. See *Paperback*, 740 F Supp at 42.

Lotus command structure. Although the opinion informs us at some length about why certain features of the Lotus interface, such as the use of "+" to represent addition, are not expressive enough to qualify for protection,[134] it does not explain its theory regarding the expressiveness of Lotus command terms such as "worksheet," and "range," or about the expressiveness found in the ordering of "range" after "worksheet" in the menu structure.[135] Indeed, the only comment the court made about why commands were ordered as they were in the Lotus interface casts doubt on the view that the arrangement was for expressive purposes. The opinion notes that the command terms were "presented in order of predicted frequency of use rather than alphabetically."[136] This suggests the ordering was for functional purposes.

The court did not inquire whether there were functional reasons for grouping certain commands together, or putting some in lower levels of the hierarchy. Nor did it ask whether there were other functional factors constraining the design of the command structure or whether other elements of the command structure were not, in fact, "original" to Lotus.[137] The court relied heavily on the fact that Paperback could have done things differently to

This reflects a view of the relationship between the program and the user interface consistent with that in *Whelan*.

It is interesting to note how much easier it is to conceptualize the relationship between other kinds of machines and their user interfaces than to conceptualize the relationship between programs and their user interfaces. Consider, for example, the relationship between the internal working parts of a wristwatch machine and its user interface. The wristwatch (which consumers tend to consider in an integrated way as consisting of the machine and its interface) often has a user interface consisting of a face with numbers on it and two hands fixed at the center of the face. These aspects of the watch's user interface not only present a visual appearance, but also display information in an efficient manner (which we refer to as "telling" us "the time"). Another part of the user interface of the watch is the externally visible device with which one can wind and set the watch to the proper hour. This controls the functioning of the internal working parts, rather than displaying information.

[134] *Paperback*, 740 F Supp at 66-67.

[135] The second line of the Lotus and Paperback command menu displays is known as a "long prompt." Id at 64. As a user moves the cursor along the first line of the menu, thereby highlighting different first line commands, the long prompt line will display different information for each highlighted command. Some long prompts are explanations of highlighted command terms. Others display the next array of command choices that are available to the user if the user cares to invoke them. Id. The explanatory long prompts are clearly more "expressive" in character than are long prompts that display further menus of command choices. Yet, *Paperback* does not differentiate between the explanatory and submenu long prompts in assessing the expressiveness of the Lotus interface. The court does, however, observe that Paperback's explanatory long prompts are different from Lotus's. Id at 70. See note 17 for other differences between the Lotus and Paperback user interfaces.

[136] *Paperback*, 740 F Supp at 67. It is difficult to believe that the Supreme Court would find an ordering based on frequency of use significantly more expressive than the alphabetic ordering found unexpressive in *Feist*, 111 S Ct 1282.

[137] Although the court recognizes that the use of "/" to invoke command menus was actually original to Dan Bricklin, a co-developer of the Visicalc program, which was the first successful electronic spreadsheet program, the court disposes of Lotus's claim to the "/" on merger ground rather than on ground that it was not original to Lotus. *Paperback*, 740 F Supp at 66. Although Lotus later acquired the copyright in Bricklin's Visicalc program, at the time Lotus 1-2-3 was developed, this feature of Visicalc was copied by the developers of Lotus 1-2-3. See Affidavit of Daniel S. Bricklin, *Lotus Development Corp. v Paperback Software Intl*, CA No 87-0076-K at 22 (June 28, 1990). The developers of Lotus 1-2-3 copied a number of other elements of the Visicalc user interface. Id at 34-36. "In most cases, Lotus 1-2-3 uses the same [command] terms as Visicalc . . . ." Id at 34. See also

support the conclusion that the Lotus command structure was "expressive."[138] That something can be done differently may be relevant to an "expressiveness" inquiry, but insufficient to demonstrate what copyright law considers to be "original expression."

One line of cases, presenting analogous copyright problems to this dispute that the court does not mention, is the "blank form" cases.[139] In these cases, judges examine the forms in question to determine whether they are expressive. When the forms contain explanatory material, courts will generally find sufficient expressiveness to support copyright protection for the form.[140] On the other hand, when the only text in the forms consists of terse instructions or descriptors identifying categories of information to be elicited when the form is used, courts will generally find insufficient "original expression" to support a copyright.[141] Even though it may require some intellectual effort to select which categories of information should appear on a form and how these categories should be arranged, and even though there may be many ways to do these things, courts still require more from a form designer before the work can be considered an "original work of authorship."[142]

It is easy to identify expressive components in highly fanciful and artistic user interfaces. In more functional interfaces, however, particularly those that

Christopher Barr, *From Visicalc to 1-2-3*, PC Magazine 169 (May 26, 1987) (showing how Lotus built on the command structure of Visicalc).

[138] See note 51 and accompanying text. Although the court recognized that some command terms in the Lotus interface were "obvious or merge[d] with the idea," it stated that this did "not preclude copyrightability for the command structure taken as a whole." *Paperback*, 740 F Supp at 67. Without quarreling with this statement, it is still fair to ask what was expressive about the command structure.

[139] See, for example, *Bibbero*, 893 F2d 1104 (medical billing forms ruled uncopyrightable for lack of conveyance of information); *John H. Harland Co. v Clarke Checks, Inc.*, 711 F2d 966 (11th Cir 1983) (denying copyright protection to checkbook format); *Janus Marketing Communications, Inc. v Doubleday & Co.*, 569 F Supp 76 (SDNY 1981) (daily activity charts ruled uncopyrightable). These cases are among many that trace the "blank forms" rule to *Baker v Selden*.

[140] See, for example, *Beardsley*, 253 F2d 702 (pamphlet containing forms with explanatory information held copyrightable).

[141] See note 137. See also *Safeguard Business Systems, Inc. v Reynolds & Reynolds Co.*, 14 USPQ2d (BNA) 1829, 1832 (ED Pa 1990) ("The Safeguard day sheets are more innovative and sophisticated than the ledger pages in *Baker v Selden*, and they do convey a certain amount of information. These forms include certain visual guides to where some numbers should be recorded, but are not sufficiently informative to be subject to copyright.").

[142] See, for example, *Bibbero*, 893 F2d 1104 (despite a considerable amount of printing on a medical form and the fact that the information could be arranged differently, the court found no copyrightable expression). The *Bibbero* court acknowledged that cases interpreting the blank forms rule "do not yield a consistent line of reasoning." Id at 1107. It distinguished one of the cases on which the plaintiff relied and stated its disagreement with the ruling in another. Id. The court noted that the Copyright Office had recently restudied the blank forms regulation, 37 CFR § 202.1(c). Despite arguments made by blank forms suppliers about their need for copyright protection, it found no persuasive argument for repealing this regulation and reaffirmed the continuing importance of *Baker v Selden* as authority for the regulation. Id. The blank form suppliers were most likely relying on the "sweat of the brow" cases which tested copyrightability by the hard work involved in compiling facts as opposed to by originality. These, however, were recently spurned by the Supreme Court in *Feist*, 111 S Ct 1282. In *Feist*, as in *Bibbero*, the Court took seriously the requirement that there be something "expressive" to support copyright protection for the work.

are command driven, it is more challenging to identify expressive elements.[143] Those who design functional interfaces, like those who design forms, generally aim to maximize system efficiency and ease of use.[144] Designers choose command names indicating the function the term represents.[145] When a program has several commands, designers limit the number of command terms displayed at one time so that users are not overwhelmed and confused with choices.[146] This necessitates the creation of a hierarchy of commands, with those of a more general character available higher up in the menu of choices, and a grouping in submenus of commands with related functions.[147] Such ordering requires intellectual effort, but it is a kind of effort that resembles designing forms, and even more so, designing other human-machine interfaces that are outside the bounds of copyright protection.[148]

One sign of how unexpressive the Lotus interface is and how little Lotus has treated the 1-2-3 command structure as part of the work's expressive authorship is the large number of books written about the Lotus program that reproduce the Lotus commands and their arrangements, and explain how the commands can be used to accomplish certain tasks.[149] If the Lotus command structure is highly expressive, it would seem unnecessary for there to be so many books explaining how to use the commands, navigate the structure, and accomplish different kinds of spreadsheet functions.[150] Lotus, who asserted against Paperback that reproducing these aspects of its interface is copyright infringement, has not sued the authors of these books, even though they have

[143] Explanatory long prompts and on-line help facilities are examples of expressive aspects of the Lotus 1-2-3 interface. Graphical design elements of other user interfaces may also be expressive enough to be protectable by copyright.

[144] See Curtis, 30 Jurimetrics J at 74 (cited in note 130) ("The importance of aesthetics relative to other user interfaces increases with the importance of playfulness and decreases with the importance of productivity.").

[145] See, for example, Gary Perlman, *Natural Artificial Languages: Low Level Processes*, 20 Intl J Man-Mach Stud 373 (1984) (discussing choices of mnemonic names for commands).

[146] See, for example, Margaret Gardner & Bruce Christie, eds, *Applying Cognitive Psychology to User Interface Design* 268-69 (1987) (guidelines for screen design and organization).

[147] See Id.

[148] The design of cockpit control panels for airplanes or of dials and switches for monitoring the safe operation of power plants also involve the arrangement of informational elements to facilitate the proper functioning of machines. Yet these kinds of human-machine interfaces are not protectable by copyright law because of the "useful article" doctrine of copyright law. See notes 164-170 and accompanying text.

[149] There are more than two hundred listings in the current issue of *Books in Print* having "Lotus 1-2-3" in the title, among them: Bill Kling, *The ABC's of Lotus 1-2-3* (Scott, Foresman, 1990); Ira Krakow, *Lotus 1-2-3: Self Taught* (Brady Communications, 1989); Que Corp., *Using 1-2-3* (Que Corp, 1987); Alan Simpson & Paul Lichtman, *The First Book of Lotus 1-2-3, Release 2.2* (H.W. Sams & Co., 1990); Mark Williams & Nelda M. Rinckenberger, *Expert Advisor 1-2-3, Release 3.2* (—, 1990); Jeff Woodward, *Teach Yourself Lotus 1-2-3, Release 2.2* (Sybex, 1990).

[150] This in itself suggests that the Lotus command structure is part of a "system" that the program's developers had devised for performing a set of spreadsheet functions, even apart from the role of the command structure in the macro facility. Whole chapters of the books listed in note 149 discuss specific commands. See, for example, Kling, *The ABC's of Lotus 1-2-3* ch 9 (cited in note 149) (lessons on moving, inserting, and deleting in 1-2-3).

reproduced more of the Lotus commands and arrangements than Paperback, which reproduced the Lotus command words only once per program.[151]

The Ninth Circuit recently ruled, in *Ashton-Tate Corp. v. Ross*,[152] that a spreadsheet program command structure was insufficiently expressive to be protectable by copyright. Ross had collaborated with a colleague to develop a spreadsheet program for the Macintosh computer. His job was to develop the "engine" for the program, and his colleague's was to design the user interface. In the process of developing the engine, Ross made a list of commands that he thought should be included in the user interface, grouping them in certain arrangements to represent the various menus the program would have. After Ross and his colleague had a falling out, the colleague went to work for Ashton-Tate and prepared a Macintosh spreadsheet program for that firm. The Ashton-Tate product has a user interface in which, Ross alleged, "[n]ot only are the individual commands identical to those of his program [but] the order in which they are displayed and the menus in which they are contained are identical to the command set . . . ."[153]

The district court ruled that Ross's "list of commands is only an idea that is not protected under federal law."[154] On appeal, Ross argued that in several other cases, the ordering and arrangement of user interface commands had been protected by copyright law. These cases emphasized the large number of different arrangements that were possible.[155] The Ninth Circuit, however,

[151] See, for example, Que Corporation Staff, *Using 1-2-3* at 496-97 (cited in note 149); Williams, *Expert Advisor 1-2-3, Release 2.2* (cited in note 149) (showing representations of Lotus menu screens). If Lotus sued the authors of these books for copyright infringement because they reproduced Lotus's commands, command structure, and mode of presenting the commands, the authors would surely defend themselves by asserting that they were merely explaining the Lotus spreadsheet system to users. Lotus is, of course, unlikely to sue these authors. Precisely because the 1-2-3 interface is so unexpressive (in a copyright sense), Lotus benefits from the efforts of authors who explain to users how to implement various spreadsheet functions by interacting with the Lotus interface. Lotus seems more concerned about protecting itself from competition in the electronic spreadsheet market than in protecting the command structure more generally.

[152] 728 F Supp 597 (ND Cal 1989), aff'd, 916 F2d 516 (9th Cir 1990).

[153] Borland Brief at 51 (cited in note 17), quoting from Ross's Brief in Opposition to Ashton-Tate's Motion for Summary Judgment at 14. Following page 51 of the Borland brief is a photocopy of the handwritten list Ross had developed.

[154] *Ross*, 728 F Supp at 602. The district court stated:

> The document given to Wigginton is only a list of labels for user commands, many of which are common commands that were already available on other software programs. There is nothing innovative or novel about the labels that Ross proposed Wigginton use for the program or the order in which they are listed on the document. The single sheet of paper does not contain any source code. The document clearly falls short of the threshold separating ideas from expressions . . . . Ross merely told Wigginton what tasks he believed the interface should allow the user to perform.

Id. After stating that the list of commands was only an idea not protected by federal law, the court cited 17 USC § 102(b). Id. See also NRC Report at 54 (cited in note 16) (expressing the doubts of some software developers that a distinction between idea and expression can be made in program user interfaces).

[155] Borland Brief at 52 (cited in note 17). The cases cited include: *Manufacturers Technologies*, 706 F Supp 984 (developer's arrangement of items in a user interface held copyrightable expression); *Softklone*, 659 F Supp 449 (status screen arrangement for computer program user interface held protected by copyright); *Broderbund*, 648 F Supp 1127 (arrangement of terms on computer screens protected).

affirmed the district court ruling on this point.[156] In the appellate court's view, there was not a triable issue of fact regarding "expressiveness" in Ross's set of user interface commands. The commands were simply the names of the functions that the program was capable of performing, grouped in a way to promote efficiency in using the program. It may require intellectual effort to identify what functions the program should perform, and to group the commands to facilitate efficient accomplishment of these tasks, but that does not make the commands "expressive" in the way required to be afforded copyright protection.[157] Thus, in light of this opinion, the existence of books explaining Lotus commands and functional factors pertaining to command structures makes the expressiveness of the Lotus command structure questionable.

B. Rethinking the Functionality of Computer Programs and User Interfaces as It Bears on the Proper Scope of Copyright Protection

Perhaps because the lawyers' arguments also obscured this issue, *Paperback* contains a muddled discussion about the functionality of computer programs and user interfaces and the corresponding copyright implications. Nowhere in the opinion is there recognition of the long-standing principle that the scope of copyright protection for functional writings is quite narrow.[158]

[156] *Ross*, 916 F2d at 521-22.

[157] One way to perceive the abstractness of spreadsheet commands is by considering them as "nonliteral" elements of a spreadsheet program. Each command can be viewed as standing at the peak of the hierarchy of abstractions for the code that will implement that particular function. For example, assume that command "Move" was associated with certain lines of the program source code. The source code instructions themselves would be the "literal text" of the copyrighted program. A detailed summary of the sequence of instructions might be a structural abstraction for those lines of code which could accurately be described as a "nonliteral" element of the program. Even more abstract (that is, less detailed) representations of this sequence might also be made which would also be nonliteral in nature. The most abstract representation of the function for that segment of the program would be the command name that would appear in the user interface. In this way of understanding the command name, it is a "nonliteral" element of the program, but one of a much more abstract character than those that might represent the underlying structure of the program. See Alfred Z. Spector, *Software, Interface, and Implementation*, 30 Jurimetrics J 79, 86 (1989) (a computer scientist's discussion of the abstractness of user interfaces).

Thus, a command term is a very abstract representation of one function of the program. The code associated with that one function could be written as a separate program. Because programs are most valuable when they permit users to perform a number of related functions, mass-marketed programs tend to consist of groups of functions. Those functions tend to be associated with a command name (or symbol) that will be displayed on a screen as part of a user interface. Viewed in this way, each command is a nonliteral element of a program, but is the kind of nonliteral element that, properly understood, is an "idea." A user interface consisting of a set of command terms is a list of ideas, as the Ninth Circuit correctly observed in *Ross*, discussed in notes 152-156 and accompanying text. Unless it is copyright infringement for a program to have the same compilation of functions as another program, user interface similarities involving arrangements of commands may more often be at the "idea" level than the "expression" level.

[158] See *LaST Frontier Report*, 30 Jurimetrics J at 18-19 (cited in note 6). The court treats Paperback's argument that the scope of protection for computer programs should be narrow as if it were a "policy argument" and consigns discussion of it to the "postscript" section of the opinion. *Paperback*, 740 F Supp at 77-79. Treating the argument this way caused the court to consider it as an argument better addressed to Congress than to the courts which were bound to follow the law as it was. The court also regarded this argument as contrary to congressional objectives of providing incentives to software innovators. The "narrow scope for functional writing" principle should

Instead, the opinion continuously refers to copyright cases involving novels, dramatic plays, and fabric designs. Courts have traditionally given broad protection in these areas because of their artistic and fanciful character.[159] Despite its concentration on these cases, the court did not regard the Lotus interface as an artistic or fanciful work. In one subsection of the opinion, the court recognizes the functional character of the Lotus interface.[160] The proper copyright consequences of this characterization, however, were not clear to the court. The opinion dismisses Paperback's functionality argument before the court sets forth the test for "copyrightability" to be used in the case.[161] The functionality of the Lotus interface is not discussed in the parts of the opinion that analyze the copyright infringement.[162]

The subsection of *Paperback* on the functionality issue begins with a reference to what was apparently one of Paperback's several functionality arguments.[163] This argument compared the Lotus user interface to other human-machine interfaces, which are outside the scope of copyright law.[164]

instead have been integrated into the body of the court's infringement analysis, for it is a well-established principle of copyright law which applied to the dispute just as clearly as it would to any other functional writing case. The CONTU Report and legislative history relevant to the scope of copyright protection for computer programs reflect congressional intent that standard principles of copyright law of this sort were to be applied in computer program cases. See CONTU Report (cited in note 5) and text accompanying note 188.

[159] *Paperback*, 740 F Supp at 51-52 (citing eight artistic and fanciful work cases for the proposition that nonliteral elements of such works can be protected by copyright law); id at 54 (concluding that protecting nonliteral elements of copyrighted computer programs was "consistent with the treatment of other kinds of intellectual works—specifically, with the treatment of nonliteral elements of expression in musical, dramatic, and motion picture works, and works of literature"); id at 60 (citing artistic or fanciful work cases in support of use of the "patterns of abstraction" test and similar standard principles of copyright law); id at 70 (comparing the similarities between the Lotus and Paperback interfaces to those in a case involving fabric designs).

[160] Id at 54-58. The court states that "Lotus 1-2-3 is surely useful." Id at 57.

[161] Id at 58-59.

[162] Id at 65-70 (identifying the copyrightable elements of the Lotus interface and finding that the copying by Paperback was sufficiently substantial to be infringing).

[163] Id at 54. The court was quite critical of Paperback's attorneys for making their functionality argument in many forms and for failing to state every premise and assumption of their argument. Id at 56.

[164] Id at 54-55. Put in its best light, Paperback's "useful article" argument might have been that the Lotus 1-2-3 interface is a human-machine interface; that human-machine interfaces have traditionally been proper subject matter for patent—not copyright—protection; that traditional copyright law would reject human-machine interfaces from its domain under the "useful article" doctrine; and that, in the Lotus interface, there was no separable "artistic" or "expressive" aspect that could be protected by Lotus's copyright in the program. Every aspect of the Lotus interface is directed to achievement of spreadsheet functionality, just as were the columns and headings of Selden's ledger sheets in *Baker v Selden*.

It is somewhat surprising that the court in *Paperback* did not reject the "useful article" defense as inapplicable because the program was registered as a "literary work." Some would say this rule only applies to pictorial, sculptural, or graphic works. See *Harper House, Inc. v Thomas Nelson, Inc.*, 889 F2d 197 (9th Cir 1989) (stating that the "useful article" doctrine does not apply to textual elements of organizers developed to assist people in planning their activities, but ruling that some aspects of the organizers were unprotectable because of their utilitarian character). The court in *Paperback*, however, seems to accept that the doctrine had some relevance in computer program cases—an aspect of his ruling quite consistent with *Baker v Selden*—because it states its understanding of the doctrine more broadly than construction of the statutory provision might suggest: "those elements of a useful article that can exist independently of the utilitarian aspects of the article are potentially copyrightable because those elements are elements of *expression* that can be distinguished from the

Two examples given were the "H" gear shift pattern for cars and the "QWERTY" keyboard arrangement. In *Synercom Technology, Inc. v. University Computing Co.*,[165] the court analogized the input formats for Synercom's statistical analysis program to the uncopyrightable "H" gearshift pattern, holding that there was no copyrightable expression in the formats that was "separable" from the idea they embodied. *Paperback* construed *Synercom* as having held "that the expression of nonliteral sequence and order is inseparable from the idea and accordingly is not copyrightable."[166] *Paperback* then cited a number of cases that had either held or accepted the principle that nonliteral elements of computer programs could be protected by copyright law to support its rejection of *Synercom*.[167]

The *Paperback* opinion made several statements that reveal the heart of the court's concern about Paperback's functionality arguments:

> If, in a context such as that of *Synercom* or of this case, an idea and its expression were taken to be inseparable and the expression therefore not copyrightable, copyright law never would, as a practical matter, provide computer programs with protection as substantial as Congress has mandated—protection designed to extend to original elements of expression however embodied.
>
> I credit the testimony of expert witnesses that the bulk of the creative work is in the conceptualization of a computer program and its user interface, rather than in its encoding, and that creating a suitable user interface is a more difficult intellectual task, requiring greater creativity, originality, and insight, than converting the user interface design into instructions to the machine.
>
> Defendants' contentions would attribute to the statute a purpose to protect only a narrowly defined segment of the development of computer programs, and to preclude from protection even more significant creative elements of the process.[168]

After construing Paperback's functionality arguments as involving "word games" about the meaning of "useful articles" in copyright law,[169] the court agreed with Paperback that the Lotus interface was useful, but insisted that

utilitarian functions of the article." *Paperback*, 740 F Supp at 52 (emphasis in the original). Yet the court does not identify what separable expressive elements exist in the Lotus interface. The court's analysis instead shifts to the *Whelan*-like idea/expression test. See notes 45-52 and accompanying text.

[165] 462 F Supp 1003 (ND Tex 1978).

[166] *Paperback*, 740 F Supp at 55.

[167] Id. Note that the court converts the functionality argument into a nonliteral similarity argument for which there are precedents to support its position. See note 90 for a further explanation of *Paperback's* misunderstanding of *Synercom*. After string-citing several "nonliteral elements" computer program cases, the court acknowledges that *Softklone*, 659 F Supp 449, and *Plains Cotton*, 807 F2d 1256, had taken a somewhat different view, although without indicating that both had cited approvingly to *Synercom*, as had *E.F. Johnson v Uniden Corp of America*, 623 F Supp 1485, 1500 (D Minn 1985); *SAS Institute, Inc. v S & H Computer Systems, Inc.*, 605 F Supp 816, 826 (MD Tenn 1985); and *Kepner-Tregoe*, 203 USPQ (BNA) at 133.

The court then asserts that the Synercom input formats were "quite different" from the nonliteral elements of Lotus 1-2-3, although without saying how or why. *Paperback*, 740 F Supp at 55.

[168] Id at 56 (citations omitted). To clarify my analysis of the court's concerns, I have separated in the text these three statements. However, they appear in the opinion as consecutive sentences in one paragraph of the opinion.

[169] Id at 56, 71-73. The court states three different forms of its understanding of Paperback's "useful article" defense. Id at 56-57. The arguments are not restated here, for they absurdly pose the issue, and not surprisingly, the court finds them so rife with flaws as scarcely to require a reasoned response.

> [i]t does not follow that when an intellectual work achieves the feat of being useful as well as expressive and original, the moment of creative triumph is also a moment of devastating financial loss—because the triumph destroys copyrightability of all expressive elements that would have been protected if only they had not contributed so much to the public interest by helping to make some article useful.[170]

The court then shifted the focus of its analysis to the idea/expression distinction as interpreted in *Whelan v. Jaslow*,[171] and no further word is heard about functionality issues. There are several noteworthy things about the functionality portion of the *Paperback* opinion. First, it contains no reference to section 102(b), *Baker v. Selden*, or any of the functional work cases, except *Synercom*, whose "central proposition" the court rejects.[172] In addition, it does not recognize that the CONTU Report explicitly mentions *Baker v. Selden* and other functional work cases as copyright precedents on the idea/expression

[170] Id at 57. The court then compares the usefulness of computer programs to that of dictionaries, directories, and maps, to illustrate that just because something is useful does not necessarily mean it is not copyrightable. Id at 58. This is an inappropriate comparison because dictionaries, directories, and maps are not considered to be "useful" in a copyright sense because their sole function is to convey information or portray an appearance. See Samuelson, 1984 Duke L J at 727-49 (cited in note 5). Computer programs are useful because they are processed in machines to perform functional tasks the same as other machines. Computer program user interfaces are just as much human-machine interfaces as are microwave oven control panels.

The court further argues: "To hold [that a work was uncopyrightable because it was associated in the marketplace with a high degree of usefulness] would be to deny copyright protection to the most original and least obvious products of the creative mind merely because the marketplace accepts them as distinctively 'functional.' Such a rule would grant copyright protection for only those products that fall far short of being the best available." Such a rule "would offer incentives to market only the second, or third, or tenth best, and to hold back the best for fear that it is too good for copyrightability." *Paperback*, 740 F Supp at 58. Near the end of *Paperback*, where the "standardization" defense is raised, the court expressed similar concerns:

> By arguing that 1-2-3 was so innovative that it occupied the field and set a *de facto* industry standard, and that, therefore, defendants were free to copy plaintiff's expression, defendants have flipped copyright on its head. Copyright protection would be perverse if it only protected mundane increments while leaving unprotected as part of the public domain those advancements that are more strikingly innovative.

Id at 79. But see note 88 for a discussion of *Baker v Selden*'s perspective on the irrelevance of the novelty in copyrighted works.

[171] The functionality subsection concludes with this statement which shows how the court shifts away from functionality to idea/expression:

> [A] court, in determining whether a particular element is copyrightable, must not allow one statutory mandate—that functionality or usefulness is not itself a basis for copyrightability—to absorb and destroy another statutory mandate—that elements of expression are copyrightable. Elements of expression, even if embodied in useful articles, are copyrightable if capable of identification and recognition independently of the functional ideas that make the article useful. This mandate may be viewed as a corollary of the central distinction of copyright law between idea and expression . . . .

Paperback, 740 F Supp at 58.

See notes 41-52 and accompanying text for a discussion of *Paperback*'s use of a *Whelan*-like analysis of idea and expression.

[172] This section contains numerous references to computer program cases, but most of which concern the "nonliteral" elements issue. One "useful article" case is cited for the proposition that a copyrighted work does not lose its protected status when subsequently put to functional use. *Brandir Intl, Inc. v Cascade Pacific Lumber Co.*, 834 F2d 1142 (2d Cir 1987). It is odd that *Paperback* relied on this case because, in it, the court ruled that a slightly modified version of the plaintiff's sculpture was unprotectable by copyright because the changes made to the sculpture were functionally related to its new function as a bicycle rack, causing it to become a "useful article."

distinction that should be considered in computer program cases.[173] Although the court recognizes the "merger" principle of copyright law, it seems unwilling to apply the principle except as to trivial details of the Lotus program.[174]

More importantly, however, the court's statements reflect fundamental misunderstandings of copyright law and principles. It is, for example, inappropriate to say that user interfaces should be protected by copyright law because the bulk of the creativity they embody resides in the "conceptualization" of them.[175] The text of section 102(b) indicates that copyright does not protect "concepts" or "conceptualizations," no matter how creative or original they are.[176] The most creative thing about Selden's book was undoubtedly the bookkeeping system explained in it, not the prose he used to describe it. Yet the Court ruled that the scope of his copyright was limited to his explanation of the system.[177]

Nor is it proper to test whether an aspect of a copyrighted work is protectable by measuring how much hard work went into either creating the idea or implementing it in some concrete form. The Supreme Court's recent *Feist* decision evidences that "sweat of the brow" does not automatically signify the presence of "original expression" protectable by copyright law.[178] Moreover, creativity at the conceptual level does not always indicate that copyrightable expressiveness will be present in all aspects of the concept's written implementation of it, as *Baker v. Selden* also illustrates.[179]

The "otherwise not enough protection" argument is also not a proper copyright argument.[180] Had such an argument been made in *Baker v. Selden*, the Court's likely response would have been "that's what patents are for."[181] Although there is some uncertainty at present about the patentability of computer program-related inventions, user interface patents are now issued. The Lotus interface, or at least some elements of it, might have been eligible for a patent.[182] Even if patent protection was unavailable for valuable aspects of the Lotus interface, the "otherwise not enough" argument may more properly be construed as an argument for some *sui generis* protection for user interface features such as command hierarchies.[183]

[173] See CONTU Report at 18-20 (cited in note 5).

[174] See *Paperback*, 740 F Supp at 66. See also notes 48-51 and accompanying text.

[175] Id at 56.

[176] 17 USC § 102(b) (language quoted at text accompanying note 3).

[177] See note 87 and accompanying text.

[178] *Baker v Selden*, 101 US at 102. See *Feist*, 111 S Ct 1282.

[179] See notes 85-92 and accompanying text.

[180] See *Feist*, 111 S Ct at 1289-90. The strongest argument for "bending" the originality requirement of copyright law to allow "sweat of the brow" protection for compilations was that without copyright protection, there would be inadequate incentives to invest in the socially desirable activity of compiling information. See Ginsburg, 90 Colum L Rev at 1899 (cited in note 108). Yet, the Supreme Court in *Feist* expressly rejected this rationale for recognizing copyright protection for these kinds of works. See *Feist*, 111 S Ct at 1289-90.

[181] See *Baker v Selden*, 101 US at 102; note 88 and accompanying text.

[182] See note 113 and accompanying text.

[183] See, for example, Stern, 14 Colum-VLA J L & Arts at 355 (cited in note 13); Abramson, 7 Computer L at 9-10 (cited in note 13).

Although the court in *Paperback* did not accept the idea that achieving an optimally useful user interface for a computer program could result in a "devastating" loss of protection,[184] this result is consistent with traditional principles of copyright law. In a report on the application of copyright principles to computer programs, ten intellectual property scholars agreed that

> copyright should not protect aspects of an interface that optimize, in a way for which there is no viable substitute, such design goals as rapid execution, accuracy of results, error reduction, number, and/or speed of keystroke functions, or time, effort, or cost of becoming skilled at using the program. Such functionally optimal aspects of an interface should not be protected, regardless of whether the original designer consciously employed systematic design analysis aimed at optimization or simply discovered an optimal interface aspect by intuition.[185]

An optimal computer program user interface would, in these scholars' views, be an instance of "idea/expression merger."[186] The conferees did not expect it would be easy to establish a "functional optimality" defense, but regarded it as consistent with copyright principles to recognize it.[187]

The principle that copyright law does not protect creative concepts, hard work, optimally efficient expressions, or other valuable elements of works failing to satisfy copyright standards may, on occasion, seem to lead to unfair results. As the Supreme Court recently observed:

> It may seem unfair that much of the fruit of a compiler's labor may be used by others without compensation. As Justice Brennan has correctly observed, however, this is not "some unforeseen byproduct of a statutory scheme." It is, rather, "the essence of copyright," and a constitutional requirement. The primary objective of copyright is not to reward the labor of authors, but "[t]o promote the Progress of Science and useful Arts." . . . To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information [or other uncopyrightable elements] conveyed by a work.[188]

The court in *Paperback* failed to recognize that *Baker v. Selden*, the other functional writing cases, and section 102(b) embody the copyright principle that the scope of copyright protection for functional writings is "thin," protecting only expressive aspects of such works, not details of their functional content. CONTU expected the courts to heed these three sources

[184] 740 F Supp at 57. Although there are places in *Paperback* where it might seem that the court regarded the Lotus interface as optimally functional (see, for example, id at 57-58, 79), there was no finding on this point. It is not asserted here that the Lotus 1-2-3 user interface was functionally optimal. The point is rather that the court in *Paperback* was unable to accept the idea that a functionally optimal interface might be unprotected by copyright law despite that it is consistent with traditional principles of copyright law to hold that a functionally optimal interface is unprotectable.

[185] *LaST Frontier Report*, 30 Jurimetrics J at 28 (cited in note 6).

[186] See id at 27-28.

[187] See id at 28. The "moment of achievement as moment of loss" argument is also clearly inconsistent with *Baker v Selden*. The moment of Selden's crowning achievement was the moment of devastating loss, for the Supreme Court ruled that by publishing his book, Selden had dedicated the useful system in the book to the public domain unless patented. *Baker v Selden*, 101 US at 104. Baker was free to copy the most important and valuable elements of the book because they were components of the useful system described in it. That the elements were "original" and "expressive of the system," that they were made up of lines on paper and words, did not change the Court's thinking about their utility or their expressiveness being inextricably interconnected with the idea.

[188] *Feist*, 111 S Ct at 1289-90 (citeation omitted).

in interpreting copyright protection for computer programs.[189] CONTU's assurances to Congress that the principles embodied in these sources would yield a proper balance of the interests of program authors, competitors, and the public are consistent with the views that led Congress to make section 102(b) a part of the statute. These principles are also not unintended byproducts of the law, but of its very essence.

VI

CONCLUSION

The copyright provision defining the term "computer program" makes clear that the set of statements or instructions to be used directly or indirectly to bring about certain results is to be protected by copyright law.[190] Neither the statutory definition, nor the legislative history of the provision, indicates that the "results" achieved when the instructions are processed in the computer are within the scope of the program copyright. Some program results, such as highly fanciful videogame graphics, may have sufficient "expressive" content that copyright will and should protect them. A user interface, or other program result should not, however, be automatically categorized as "expressive" merely because it is a result, because it is valuable, or because hard work or creativity was required to bring it into being. "Results" of computer programs are often functional in nature and beyond the scope of copyright protection.[191]

Whether an aspect of a copyrighted work should be protected by copyright should be tested by long-standing principles of copyright law requiring that the aspect be "expressive." The court in *Paperback* did not convincingly explain what was expressive about the aspects of the Lotus interface as to which infringement was found. In particular, it neglected to consider the implications of its indications that an important element of the Lotus interface, namely, the Lotus command structure, was a constituent component of the Lotus macrocommand system. The court also failed to inquire whether other aspects of the Lotus interface were constituent parts of the Lotus program's system for managing spreadsheet functions, which others needed to be able to copy in order to "express" the same system.

This article raises is not asserting that there was nothing "expressive" about the Lotus 1-2-3 user interface. The text of the opening screens, the explanatory text in the on-line help facility, and the explanatory long prompts seem to satisfy copyright standards for what constitutes original expression.

[189] See CONTU Report at 18-23 (cited in note 5).

[190] 17 USC § 101 (West Supp 1991) (definition of "computer program").

[191] A recent article, for example, reports that advances in computer modeling and tools for analysis of data from materials research labs have led to significant advances in the development of materials such as "extra-tough steel . . . for bearings in space shuttle[s] . . . ." Otis Port, *The New Alchemy*, Bus Wk 48 (July 29, 1991). These new materials may in a real sense be "results" of processing programs in computers, but it would be improper to consider them derivative works within the scope of the programmer's copyright. See generally Pamela Samuelson, *Allocating Ownership Rights in Computer-Generated Works*, 47 U Pitt L Rev 1185 (1986).

Paperback, however, did not copy these aspects of the Lotus interface,[192] which is why this article questions the court's holding.[193] Whether there were other expressive aspects of the Lotus interface that Paperback did copy, apart from those that were necessary to copy in order to "express" the same spreadsheet management or macro language system, has yet to be determined.

While this article doubts about whether there were other expressive elements copied by Paperback, it is a subject on which reasonable minds might disagree. It may be that upon appropriate detailed findings and a proper copyright analysis, the court might still have found that Paperback infringed the Lotus copyright. Paperback did copy more of the Lotus interface than just the 1-2-3 command structure. It also copied much of the mode of presentation of the Lotus commands. When the Paperback program was in operation, an onlooker could have thought he or she was observing the Lotus program in operation.[194] Exact or near-exact copying of a functional writing can be a copyright infringement. A ruling of this sort in *Paperback* would have been consistent with traditional principles of copyright law. It could also have provided significant guidance to the software industry about what aspects of user interfaces could properly be protected by copyright law and why. The overbroad ruling in *Paperback* continues rather than resolves the controversy about the protectability of user interfaces by copyright law. It is unfortunate that only by many more years of litigation can the software industry get definitive guidance on these important issues.

[192] See note 17.

[193] See, for example, *Kepner-Tregoe*, 203 USPQ (BNA) at 134 (only minor changes in wording necessary to avoid infringement in functional writing case).

[194] See *Paperback*, 740 F Supp at 70. Note, however, that this would import a trademark-like "confusing similarity" standard into copyright law which generally looks to substantial similarity as to protected expression.

One of Lotus's expert witnesses offered a definition of "clones" of software products which the Paperback product may have satisfied. See Affidavit of Bernard Galler, *Lotus Development Corp. v Paperback Software Intl*, CA No 87-0076-K, ¶ 91. Had the court ruled against Paperback because it was a "clone" and defined "clone" with precision, it would have given guidance to the software industry. It was to prevent "slavish" imitations by competitors that courts developed the "sweat of the brow" theory of originality that the Supreme Court recently rejected in *Feist*, 111 S Ct at 1282. The validity of Lotus's copyright in the 1-2-3 program has not been questioned; still, it is questionable whether a "slavish" imitation of the Lotus product could be justified if the only aspects of the interface copied were unprotectable by copyright.