

1995 WL 728487 (U.S.) (Appellate Brief)
United States Supreme Court Amicus Brief.

LOTUS DEVELOPMENT CORPORATION, Petitioner,
v.
BORLAND INTERNATIONAL, INC., Respondent.

No. 94-2003.
October Term, 1995.
December 8, 1995.

On Writ of Certiorari to the United States Court of Appeals for the First Circuit

**BRIEF AMICI CURIAE OF AMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS AND
COMPUTER & COMMUNICATIONS INDUSTRY ASSOCIATION IN SUPPORT OF RESPONDENT**

EDWARD J. BLACK
COMPUTER & COMMUNICATIONS
INDUSTRY ASSOC.
666 Eleventh Street, N.W.
Washington, DC 20001
(202) 783-0070

PETER M.C. CHOY
Counsel of Record
AMERICAN COMMITTEE FOR
INTEROPERABLE SYSTEMS
2550 Garcia Avenue
Mountain View, CA 94043
(415) 336-2482

PAUL GOLDSTEIN
STANFORD LAW SCHOOL
Stanford, CA 94305
(415) 723-0313
Attorneys for Amici

***i TABLE OF CONTENTS**

TABLE OF CONTENTS	i
TABLE OF AUTHORITIES	iii
INTEREST OF AMICI	1
SUMMARY OF ARGUMENT	3
ARGUMENT	4
I. THE FIRST CIRCUIT'S DECISION FALLS SQUARELY WITHIN THE MAINSTREAM OF RECENT SOFTWARE COPYRIGHT DECISIONS, AND THUS IS CONSISTENT WITH THIS COURT'S DECISION IN FEIST	7
II. LOTUS MISREADS THE FIRST CIRCUIT'S OPINION	12

III. THE FIRST CIRCUIT CORRECTLY CONCLUDED THAT COPYRIGHT PROTECTION DOES NOT EXTEND TO THE LOTUS 1-2-3 COMMAND STRUCTURE	15
A. Lotus Understates the Functionality of the 1-2-3 Command Structure	16
B. The First Circuit Correctly Recognized that Compatibility Considerations Effectively Limited the Alternatives Available to Borland	20
C. Judge Boudin Correctly Identified the Adverse Consequences of Extending Copyright Protection to the 1-2-3 Command Structure	23
*ii D. Lotus and Its Amici Grossly Exaggerate the Adverse Consequences, If Any, of the First Circuit's Decision	25
CONCLUSION	29

***iii TABLE OF AUTHORITIES**

CASES

In re Alappat, 33 F.3d 1526 (Fed. Cir. 1994)	28
Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), cert. denied, 115 S. Ct. 1176 (1995)	11
Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), cert. dismissed, 464 U.S. 1033 (1984)	20
Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992)	11, 21
Baker v. Selden, 101 U.S. 99 (1879)	3, 11
Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1989)	11
Campbell v. Acuff-Rose Music, Inc., 114 S. Ct. 1164 (1994)	25
Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2nd Cir. 1992)	passim
Diamond v. Diehr, 450 U.S. 175 (1981)	28
Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994), opinion supplemented on denial of reh'g en banc, 46 F.3d 408 (5th Cir. 1995)	11
Engineering Dynamics, Inc. v. Structural Software, Inc., 46 F.3d 408 (5th Cir. 1995)	22
Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340 (1991)	passim
*iv Gates Rubber Co. v. Bando American, Inc., 798 F. Supp. 1499 (D. Colo. 1992), aff'd in part, vacated in part, 9 F.3d 823 (10th Cir. 1993)	10
Gates Rubber Co. v. Bando Chem. Indus. Ltd., 9 F.3d 823 (10th Cir. 1993)	11
Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass. 1992), rev'd, 49 F.3d 807 (1st Cir.), and cert. granted, 116 S. Ct. 39 (1995)	22
Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 223 (D. Mass. 1993), rev'd, 49 F.3d 807 (1st Cir.), and cert. granted, 116 S. Ct. 39 (1995)	18
Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir.), cert. granted, 116 S. Ct. 39 (1995)	passim
Morrissey v. Procter & Gamble Co., 379 F.2d 675 (1st Cir. 1967)	13
Qualitex Co. v. Jacobson Prod. Co., 115 S. Ct. 1300 (1995)	21
Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256 (5th Cir.), cert. denied, 484 U.S. 821 (1987)	8
Sega Enter. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992)	11, 21
United States v. Microsoft Corp., 56 F.3d 1448 (D.C. Cir. 1995)	28
Whelan Associates, Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987)	passim

***v STATUTES & REGULATIONS**

17 U.S.C. § 101	5
17 U.S.C. § 102(a)	29
17 U.S.C. § 102(b)	passim
35 U.S.C. § 102	29
35 U.S.C. § 103	29
35 U.S.C. § 112	29
60 Fed. Reg. 28,778 (1995)	28

OTHER AUTHORITIES

J. Band and M. Katoh, Interfaces on Trial (1995)	5, 21
--	-------

R. Frank and P. Cook, The Winner-Take-All Society (1995)	28
M. Knell, Microsoft Fights for Intuit, The Boston Herald, Apr. 28, 1995	18
P. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 Stan. L. Rev. 1045 (1989)	24
C. Morris and C. Ferguson, How Architecture Wins Technology Wars, Harv. Bus. Rev. (Mar./Apr. 1993)	18
M. Nimmer and D. Nimmer, Nimmer On Copyright (1995)	8
President's Information Infrastructure Task Force, Global Information Infrastructure: Agenda for Cooperation (U.S. Government Printing Office, Washington, D.C., Feb. 1995) .	24
*vi G. Saloner, Economic Issues in Computer Interface Standardization, 1 Econ. Innov. New. Tech. 135 (1990)	27
J. Sandberg and G. Hill, Microsoft Probe Spurs Subpoenas Tied to Internet, Wall St. J., Dec. 4, 1995	7
J. Sandberg, Sun and Netscape Are Forming Alliance Against Microsoft on Internet Standard, Wall St. J., Dec. 4, 1995	7
U.S. Congress, Office of Technology Assessment, Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change, OTA-TCT-527 (U.S. Government Printing Office, Washington, D.C., May 1992)	6, 19
F. Warren-Boulton, K. Baseman & G. Woroch, Copyright Protection of Software Can Make Economic Sense, 12 Computer Law. 10 (Feb. 1995)	27

***1 BRIEF AMICI CURIAE OF AMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS AND
COMPUTER & COMMUNICATIONS INDUSTRY ASSOCIATION IN SUPPORT OF RESPONDENT**

The American Committee for Interoperable Systems and the Computer & Communications Industry Association submit this brief as amici curiae and respectfully request that the Court affirm the decision below. The letters of consent of Petitioner and Respondent are being filed with this brief.

INTEREST OF AMICI

The American Committee for Interoperable Systems (“ACIS”) is an informal organization of companies that develop innovative software and hardware products that interoperate with computer systems developed by other companies.¹ Computer & Communications Industry Association (“CCIA”) members participate in many sectors of the computer *2 and telecommunications industry and range in size from small entrepreneurial firms to the largest in the industry.² Collectively, ACIS and CCIA members generate revenues in excess of \$180 billion and employ over one million people.

ACIS and CCIA believe that computer programs deserve effective intellectual property protection to give developers sufficient incentive to create new programs. At the same time, ACIS and CCIA are concerned that the improper extension of copyright law will impede innovation and inhibit fair competition in the computer industry. ACIS and CCIA seek the application of legal standards that will effectuate copyright law's fundamental aims by ensuring authors “the right to their original expression,” but also encouraging competitors “to build freely upon the ideas and information conveyed by a [copyrighted] work.” Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 349-50 (1991).

Neither ACIS, CCIA, nor their members have a direct financial interest in the outcome of this litigation. However, reversal of the First Circuit's decision would have serious anti-competitive consequences for ACIS and CCIA members and the computer industry as a whole. Specifically, extending copyright protection to elements necessary for interoperability would inhibit the ability of ACIS and CCIA companies to develop innovative, competitive products. ACIS and CCIA are filing this brief in order to address the *3 important intellectual property and competition law policy issues at stake in this case.

SUMMARY OF ARGUMENT

The decisive issue in this case is whether copyright law can protect the rules that enable two elements of a computer system to work together. To be sure, Lotus and its amici talk about other matters: the protectability of a computer program's user interface; the protectability of source code; and the claimed methodological flaws in the First Circuit decision. But, boiled down to its essence, the only question for decision by this Court is whether copyright can protect the rules that enable one computer program to connect with another.

This brief first addresses Lotus' mistaken claim that, at odds with other circuits, the First Circuit fashioned a special copyright rule for software. In fact, the First Circuit followed the rule applied in other circuits -- a rule mandated by this Court for over a century, from *Baker v. Selden*, 101 U.S. 99 (1879) to *Feist* -- that requires a thinner scope of copyright protection for utilitarian literary works than for novels and plays. The brief next rebuts Lotus' assertion that the First Circuit found separable expression in its command structure, but withheld protection from this expression because it was included in an unprotectable method of operation. Lotus misreads the First Circuit's decision. The First Circuit found no separable expression; rather, it held that the expression had merged with the method of operation.

The brief then turns to the decisive issue in this case: the protectability of interface specifications, and not the protectability of the great bulk of program elements that, under the First Circuit's decision and this Court's rulings, will continue to be protected by copyright. The 1-2-3 command structure is more than a user interface; it is the interface *4 between the Lotus program and programs -- referred to as "macros" -- that are written by users at their own considerable expense for execution in connection with the 1-2-3 program. Because the 1-2-3 command structure provides the template for the macros and because the macros are the key to compatibility, the First Circuit, consistent with holdings in other circuits, ruled that those elements necessary to macro compatibility are not protected by copyright. This ruling will not harm innovation; indeed, by promoting competition, in products that differ -- transform -- the original, it will lead to increased innovation.

ARGUMENT

Unlike traditional literary works such as novels and plays that stand alone and do not need to interact with any other work, computer programs never function alone; they function only by interacting with the computer environment in which their developers place them. This environment is absolutely unforgiving. Unless the computer program conforms to the precise rules for interacting with the other elements of the system, no interaction between the program and the system is possible. As a consequence, no matter how much better or cheaper the new program is, it will not enjoy a single sale if it cannot interoperate in its intended environment. If the developer of one part of the environment can use copyright law to prevent other developers from writing programs that conform to the system of rules governing interaction within the environment -- interface specifications, in computer parlance -- the first developer could gain a patent-like monopoly over the system without ever subjecting *5 it to the rigorous scrutiny of a patent examination. Lotus seeks to use copyright in exactly this manner.³

Neither the Lotus Brief to this Court nor the briefs of its two amici⁴ explain what is really at stake in this case, or why Lotus' position has so little support in the software industry. Indeed, the briefs of Lotus and its amici leave the misimpression that the few companies supporting the Petitioner do so out of concern that the First Circuit's decision will undermine their ability to keep others from copying their code, i.e., the actual lines of "instructions" used "in a computer to bring about a certain result." 17 U.S.C. § 101 (definition of computer program); see, e.g., AIPLA Brief at 13; Lotus Brief at 49.

ACIS and CCIA members rely on the copyright law to prevent unauthorized copying of the protectable expression contained in their code. Had the First Circuit's opinion called into question the protection of code, Lotus would have support from far more than the four companies that have filed a brief on its behalf. In truth, there is no expectation in the industry that the First Circuit's decision will endanger the scope of copyright protection for code in any respect.

*6 Lotus and its few amici press this case not out of concern for code copying, but rather in an effort to extend the copyright law to prevent the emergence of compatible products that compete with their own. This case is about whether competitors can introduce compatible products that emulate, as they must, interface specifications, the rules that form the “external design” of a program.⁵ This case is an attempt by a few companies who are the “first comers” to particular markets to use copyright law to preclude competitors from using the same external design, even though it is implemented in wholly original program code.

ACIS and CCIA believe such compatibility, or interoperability as it is sometimes called, is beneficial to both the industry and to the public. A program is compatible or interoperable with another program when it can interact with other programs in the same way as the first program. Borland's Quattro Pro program is compatible with Lotus 1-2-3 because it interacts with other programs in the same way -- both programs can run “on top of” the same operating system (MS DOS) and both programs execute the same application macros written by users and other third-parties.

Compatibility is achieved by conforming to the rules that the developer establishes as a “socket” to enable another program to “plug into” it. Extending copyright to such rules has economic consequences far broader than what was contemplated by Congress in protecting the instructions used in a computer to bring about a certain result. In concrete terms, reversal of the First Circuit's decision would threaten a large and vibrant sector of the U.S. software industry that depends *7 on its ability to interoperate with computer systems developed by other firms. Copyright protection for interface specifications would lead to monopolies within each product market in the industry. Indeed, unprecedented economic concentration would result if one firm succeeds in asserting proprietary control over a critical interface specification in the information superhighway.⁶

I. THE FIRST CIRCUIT'S DECISION FALLS SQUARELY WITHIN THE MAINSTREAM OF RECENT SOFTWARE COPYRIGHT DECISIONS, AND THUS IS CONSISTENT WITH THIS COURT'S DECISION IN FEIST.

Lotus devotes much of its argument to proving the obvious: that computer programs are literary works under the Copyright Act and, as such, are governed by the same principles as other literary works. Lotus suggests that by considering the 1-2-3 command structure's utilitarian nature, the First Circuit somehow departed from congressional intent and from precedent in other circuits. Lotus Brief at 18. Lotus fails to mention, however, that although all literary works are governed by the same principle -- specifically the idea/expression distinction embodied in Section 102(b) of the Copyright Act -- this principle dictates that utilitarian works receive a thinner scope of protection than works such as novels or plays. Indeed, this Court explicitly recognized this point in Feist.

The appellate software copyright decisions since Feist all acknowledge that utilitarian works enjoy only a thin scope of protection. Yet, prior to Feist, the courts headed in the *8 opposite direction. Dicta in one of the earlier decisions, *Whelan Associates, Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987), signaled extremely broad protection. The Whelan court defined the unprotectable idea in the program at issue to be “to run a dental laboratory in an efficient way,” 797 F.2d at 1238 n.34, while the protectable expression was “the manner in which the program operates.” *Id.* at 1239. Subsequent courts and commentators interpreted the Whelan dicta as drawing the line between idea and expression at a very high level of abstraction, suggesting the availability of copyright protection for virtually everything in a computer program except the most general statement of its function.⁷ Indeed, some courts and commentators refer to a “Whelan rule” that a computer program contains but one idea.⁸ *Id.*

Courts in other circuits quickly rejected the approach taken in Whelan's dicta. The Court of Appeals for the Fifth Circuit recognized that although computer programs are literary works, they are utilitarian and thus are strictly constrained by factors external to the program, such as industry standards. *Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc.*, 807 F.2d 1256, 1262 (5th Cir.), cert. denied, 484 U.S. 821 (1987). Accordingly, the Plains Cotton court explicitly “decline[d] to embrace Whelan.” *Id.*

The rejection of Whelan accelerated after the Feist decision. Thus, the Court of Appeals for the Second Circuit in **9 Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2nd Cir. 1992) observed that “Whelan's general formulation that a program's overall purpose equates with the program's idea is descriptively inadequate.”

The Second Circuit also challenged the two justifications the Whelan opinion had given for drawing the idea/expression line at a high level of abstraction. First, the Whelan court had observed that computer programs are “literary works,” and that copyright in other types of literary works is infringed if competitors use the works' structure and organization. The Altai court acknowledged that computer programs were “literary works,” but at the same time recognized the “essentially utilitarian nature of a computer program,” and emphasized that, “compared to aesthetic works, computer programs hover even more closely to the elusive boundary line described in Section 102(b).” *Id.* at 704.

Second, the Whelan court had noted that “among the more significant costs in computer programming are those attributable to developing the structure and logic of the program” and that protecting the structure and logic “would provide the proper incentive for programmers by protecting their most valuable efforts” Whelan, 797 F.2d at 1237. The Altai court rejected Whelan's incentive-based justification for broad copyright protection as having “a corrosive effect on certain fundamental tenets of copyright doctrine.” Altai, 982 F.2d at 712. The Second Circuit explained that

[t]he interest of the copyright law is not in simply conferring a monopoly on industrious persons, but in advancing the public welfare through rewarding artistic creativity, in a manner that permits the free use and development of non-protectable ideas and processes.

**10 Id.* at 711.

Significantly, the two foregoing points flowed directly from this Court's decision in Feist. With respect to the argument that a computer program is a literary work, the Court made clear that merely calling a work a “literary work” does not determine the scope of its copyright protection and that not all literary works are entitled to the same scope of protection: “the copyright in a factual compilation is thin”; “[t]his Court has long recognized that the fact/expression dichotomy limits severely the scope of protection in fact-based works.” Feist, 499 U.S. at 349, 350. Similarly, with respect to the incentive-based justification, “Feist teaches that substantial effort alone cannot confer copyright status on an otherwise uncopyrightable work Thus, Feist implicitly undercuts the Whelan [incentive based] rationale” Altai, 982 F.2d at 711.

Whelan's dicta had led some lower courts to conclude wrongly that “the idea-expression distinction is somewhat diluted in analysis applied in infringement cases concerning computer programs.” *Gates Rubber Co. v. Bando American, Inc.*, 798 F. Supp. 1499, 1510 (D. Colo. 1992), *aff'd in part, vacated in part*, 9 F.3d 823 (10th Cir. 1993). In contrast, the Altai court correctly recognized that because “[t]he essentially utilitarian nature of a computer program ... complicates the task of distilling its idea from its expression,” Altai, 982 F.2d at 704, the idea/expression dichotomy must be applied in software copyright cases with great rigor and extreme care.

Every appellate software copyright decision since Altai agrees with its fundamental teachings: although computer programs are literary works protected by copyright, and often require great expense and effort to develop, as utilitarian works they are entitled to “thinner” protection than novels or plays; and to ensure that they do not receive too much **11* protection, the courts must closely scrutinize the works at issue and apply the idea/expression dichotomy meticulously. See, e.g., *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992); *Apple Computer, Inc. v. Microsoft Corp.*, 25 F.3d 1435, 1439 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995); *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1347 n.12 (5th Cir. 1994), *opinion supplemented on denial of reh'g en banc*, 46 F.3d 408 (5th Cir. 1995); *Gates Rubber Co. v. Bando Chemical Indus. Ltd.*, 9 F.3d 823, 836 (10th Cir. 1993); *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992).

Underlying these appellate decisions is not only this Court's decision in *Feist*, but a long line of Supreme Court decisions cautioning against the improper extension of the patent laws via copyright law or state law. As the Court stated in *Baker v. Selden*, 101 U.S. 99, 102 (1879), “[t]o give 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.” More recently, this Court in *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 146 (1989), condemned state law protection for inventions left by the Patent Act in the public domain: “the Federal patent laws have embodied a careful balance between the need to promote innovation and the recognition that imitation and refinement through imitation are both necessary to invention itself and the very lifeblood of a competitive society.” The software copyright appellate decisions, by refusing to treat software functionality as protected expression, avoid upsetting this careful balance. See, e.g., *Atari*, 975 F.2d at 839.

The First Circuit's decision falls squarely within this body of case law. Judge Boudin's concurring opinion clearly recognized that software, although a literary work, receives a different scope of protection from novels, plays, and films: *12 “Utility did not bar copyright (dictionaries may be copyrighted), but it alters the calculus.” *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 820 (1st Cir.), cert. granted, 116 S. Ct. 39 (1995). Further, Judge Boudin does not “erect special judicial barriers to protection” of software. DEC Motion for Leave to File Amicus Curiae Brief at 3. Rather, like the other circuit court decisions cited above, he believed that Congress intended courts to apply Section 102(b) of the Copyright Act in a case-by-case manner, and the thinner protection for software, like the thinner protection for other utilitarian literary works such as dictionaries, flows directly from Section 102(b). He also recognized that failure to apply Section 102(b) strictly in software copyright cases would cause copyright to infringe the jurisdiction of the patent laws.

Further, both Judge Stahl's and Judge Boudin's opinions reflect a thorough consideration of the program element at issue and its relationship to its environment, in an effort to understand on which side of the idea/expression divide it should fall. This willingness to grapple with complex technology, rather than succumb to the temptation of superficial understanding, makes the First Circuit's decision a model other courts should follow.

II. LOTUS MISREADS THE FIRST CIRCUIT'S OPINION.

In addition to faulting the First Circuit for finding that copyright accords software thinner protection than novels and plays, Lotus attacks the First Circuit's application of Section 102(b) to the facts of this case. Lotus contends that the First Circuit agreed with the district court below that the Lotus 1-2-3 command structure contained expression separable from its ideas and functionality, but that the First Circuit nonetheless refused to extend copyright protection to this expression because it was part of the program's method of operation. The First Circuit committed this error, Lotus *13 claims, by relying on a flawed definition of “method of operation,” which led it to take a “dangerous and improper shortcut in applying § 102(b).” Lotus Brief at 18; AIPLA Brief at 7.

The linchpin of this argument is Lotus' view that the First Circuit conceded that the command structure contained separable expression. In fact, the First Circuit made no such concession. To be sure, it stated that “the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms” 49 F.3d at 816. But the First Circuit made it quite clear that these expressive choices had merged with the command structure. Thus, two sentences after its reference to Lotus' expressive choices, the First Circuit stated that “[i]f specific words are essential to operating something, then they are part of a ‘method of operation’ and, as such, are unprotectable.” *Id.* (emphasis supplied).

Similarly, in its discussion of the VCR analogy, the First Circuit stated that “the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves.” *Id.* at 817. Pursuing this thought, the court stated that “[u]nlike the labels on a VCR's buttons, which merely make operating a VCR easier by indicating the buttons' functions, the Lotus menu commands are essential to operating Lotus 1-2-3.” *Id.* (emphasis supplied). The First Circuit underscored its conclusion that the developers' expressive choices had merged with the command structure, the method of operating the program, by citing the First Circuit's venerable decision in *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675,

678-79 (1st Cir. 1967), for the proposition that “[w]hen there are a limited number of ways to express an idea ... the expression ‘merges’ with the idea and becomes uncopyrightable.” 49 F.3d at 818 n.13 (emphasis supplied).

***14** Because it concluded that the expressive terms had merged with the command structure, the question before the First Circuit -- and the real question before this Court -- was whether the command structure itself was protected expression or unprotected idea.⁹ Lotus and its amici suggest that by not applying the Altai abstraction-filtration-comparison test, the First Circuit failed to assess properly whether the command structure should receive copyright protection. See Lotus Brief at 40; AIPLA Brief at 7. This argument, too, is predicated on a misreading of the First Circuit's opinion.

The First Circuit reasoned that the Altai test might be “misleading” if applied here because it could cause the identification of expression at a low level of abstraction while obscuring the fact that the expression may be an essential part of the method of operation at a higher level of abstraction. The First Circuit's reasoning is sound where, as here, there is verbatim copying of a specific element; further abstraction would simply reveal similarities at progressively more detailed levels, while deconstructing the overall system into which the sub-elements had merged. In this case, one would end up with hundreds of identical terms, without any sense that those terms were an essential part of a larger whole.

Nonetheless, it is important to note that the First Circuit did not dispense with all three steps of the Altai test, just the first one -- abstraction. It still performed the critical second step -- filtration -- on the command structure. As the next ***15** section of the brief explains, the First Circuit performed the filtration step properly.

III. THE FIRST CIRCUIT CORRECTLY CONCLUDED THAT COPYRIGHT PROTECTION DOES NOT EXTEND TO THE LOTUS 1-2-3 COMMAND STRUCTURE.

Lotus properly states that “Section 102(b) requires courts to attempt to separate the purely functional attributes of computer programs from the particular expression chosen by the programmer to accomplish or to provide that functionality.” Lotus Brief at 38. A few sentences later, Lotus restates this proposition as follows: “the expression is protected by copyright if it is only one of numerous possibilities for providing the same functionality.” Id. at 39.

These two statements get right to the heart of the matter: Lotus and the First Circuit have different meanings for the phrase “same functionality.” For Lotus, a command structure with the “same functionality” as the 1-2-3 command structure is one that offers “exactly the same selection of functions as does 1-2-3” but with different terms in a different sequence. Id. at 41. For the First Circuit, a command structure with the “same functionality” as the 1-2-3 command structure is one that is completely 1-2-3 compatible. If Lotus' definition of “same functionality” is correct, then the 1-2-3 command structure is but one of many ways of expressing the same functionality. Conversely, if the First Circuit's definition of “same functionality” is correct, then the 1-2-3 command structure is essential to achieving the same functionality. Thus, the core inquiry before this Court, and the issue on which this case turns, is whether compatibility is a functional characteristic. Below, we will explain why compatibility unquestionably is a functional characteristic unprotected by copyright. But before we do so, we must explain the nature of the product at issue.

***16 A. Lotus Understates the Functionality of the 1-2-3 Command Structure.**

Most plaintiffs in copyright infringement cases tend to exaggerate the significance of their work in an effort to magnify the work's expressive quality. Here, by contrast, Lotus down plays the nature of its command structure in an effort to minimize the structure's inherent functionality. Throughout their briefs, Lotus and its amici repeatedly refer to the 1-2-3 command structure as a “user interface” which “communicate[s] the product's underlying functionality to users in a clearly organized presentation.” Lotus Brief at 9 (citations omitted).

The command structure, however, has a second function which Lotus relegates to footnotes but which in fact is critical to both the Lotus product and this case: it functions as the interface between the spreadsheet program and “macros,” which are

programs written by users with the 1-2-3 commands. It is this program-to-program interface function that is of primary concern to ACIS and CCIA.

A consideration of the various kinds of programs that run on modern computers will clarify this function of the command structure. All computers must run an "operating system" program, which is the fundamental program that instructs the computer how to operate, how to manage its resources, and how to run other programs. *Altai*, 982 F.2d at 698. Common operating systems include Windows 95 for IBM-compatible personal computers and UNIX for work-stations. The operating system serves as a "platform" for "application programs" such as word processing programs, database programs, multimedia games, or, as in this case, spreadsheet programs. Application programs must conform precisely to the functional requirements -- the interface specification -- of the operating system, or else the applications will not work properly.

***17** Although Lotus 1-2-3, taken as a whole, is an application program, it assumes some of the characteristics of an operating system with respect to the user written application programs -- the macros -- that attach to it. In other words, Lotus 1-2-3 serves as a platform upon which the macros run. The command structure defines the rules for interaction between the 1-2-3 macros and the Lotus platform, and thus acts as an interface specification. Put differently, the command structure provides the syntax and semantics of the communication between a 1-2-3 macro and the Lotus platform.

Many Lotus users have invested substantial time and resources developing libraries of highly complex customized macros appropriate to their business needs. Indeed, the Lotus users collectively have invested far more resources in the development of their macros than Lotus ever invested in the development of Lotus 1-2-3. Lotus claims that its programmers "spent hundreds of hours over a period of many months" developing the 1-2-3 command structure. Lotus Brief at 9. Lotus users, in contrast, collectively have spent thousands, if not millions, of hours over many years writing their macros. Because of their investment in the macros, the Lotus users are "locked-in" to the Lotus environment: as they expand their operations, they simply will not purchase spreadsheet programs developed by a Lotus competitor such as Borland unless the Borland spreadsheet can also execute their macros.

The most basic form of macro compatibility requires the Borland platform to have the ability to translate the macro's instructions into instructions intelligible to the Borland platform, and vice versa. Because the set of instructions used by the macro is a subset of Lotus' commands, the Borland platform had to translate those instructions from the macros by means of a file that exactly replicated the Lotus 1-2-3 command structure's syntax and semantics. Borland calls ***18** this file the "Key Reader."¹⁰ *Lotus v. Borland*, 49 F.3d at 811-12.

Complete macro compatibility, however, requires more than just the ability to run existing macros. Users of the Borland platform must be able to correct errors in the macros, and modify the macros to address new problems. Because the macros are constructed out of Lotus commands, the user of the Borland platform can debug or modify the macros only if the platform can display the commands on the terminal and only if the platform can understand the Lotus commands keyed in by the user. Thus, to perform the debugging and modification functions essential for complete macro compatibility, the Borland platform must reproduce the Lotus command structure. *Id.* at 811 n.3.

***19** Programmers view an interface specification such as the 1-2-3 command structure as part of the external design of a program. As the Office of Technology Assessment noted,

[p]rograms have an external design or interface -- the conventions for communication between the program and the user or other programs. The external design is conceptually separate from the program code that implements the interface (the internal design). It specifies the interactions between the program and the user or other programs, but not how the program does the required computations. There are typically many different ways of writing a program to implement the same interface.

OTA Report at 26. Using copyright jargon, it can accurately be said that the interface specification is the “system” or “method of operation” that is “expressed” by the program code.

In this case, the 1-2-3 command structure was the interface specification for the Lotus spreadsheet. Lotus designed its program so that macros had to be written with the terms and sequence of the command structure in order to operate with the 1-2-3 program. Borland subsequently entered the market seeking to compete against Lotus for the business of users who had already written macros which used the command structure as an interface specification. In order to run these third party macros, Borland adopted the same interface specification (the command structure) but implemented that specification in its own wholly original code.

***20 B. The First Circuit Correctly Recognized that Compatibility Considerations Effectively Limited the Alternatives Available to Borland.**

Unlike the district court below, the First Circuit appreciated the importance of macro compatibility, and this appreciation strongly influenced the First Circuit's perception of the command structure: “That the Lotus menu command hierarchy is a ‘method of operation’ becomes clearer when one considers program compatibility.” 49 F.3d at 817. After observing that a user-written 1-2-3 macro could not operate with an incompatible spreadsheet program, the First Circuit concluded: “As the Lotus menu command hierarchy serves as the basis for Lotus 1-2-3 macros, the Lotus menu command hierarchy is a ‘method of operation.’” Id. at 818.

Lotus nonetheless contends that Borland sought to be compatible with the 1-2-3 macros “for commercial reasons, not as a technical necessity.” Lotus Brief at 12. This argument echoes the suggestion of the Court of Appeals for the Third Circuit over a decade ago that compatibility was a “commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged.” *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983), cert. dismissed, 464 U.S. 1033 (1984).

More recently, however, other circuits have replaced “metaphysical distinctions” with “practical considerations.” *Altai*, 982 F.2d at 706. They understand that software copyright issues do not arise in a vacuum, but in a particular commercial context that as a practical matter restricts a programmer's range of possible expression. In other words, these courts have acknowledged that a work's commercial *21 context helps define the boundary between idea and expression.¹¹ Thus, the Second Circuit in *Altai* recognized that

a programmer's freedom of design choice is often circumscribed by extrinsic considerations such as (1) the mechanical specifications of the computer on which a particular program is intended to run; (2) compatibility requirements of other programs with which a program is designed to operate in conjunction; (3) computer manufacturers' design standards; (4) demands of the industry being serviced; and (5) widely accepted programming practices within the computer industry.

982 F.2d at 709-10 (emphasis supplied, citations omitted). The Second Circuit ruled that copyright does not protect program elements dictated by such external factors, including compatibility. Id.

The Federal Circuit in *Atari*, citing *Altai*, stated that “[t]he court must filter out as unprotectable ... expression dictated by external factors (like the computer's mechanical specifications, compatibility with other programs, and demands of the industry served by the program)” 975 F.2d at 839. Likewise, the Ninth Circuit in *Sega* expressly recognized that computer programs “contain many logical, structural, and visual display elements that are dictated by ... external factors such as compatibility requirements and *22 industry demands. In some circumstances, even the exact set of commands used by the programmer is deemed functional rather than creative for purposes of copyright.” 977 F.2d at 1524 (citations omitted).¹²

Lotus v. Borland presents one of those circumstances in which “the exact set of commands used by the programmer is deemed functional” because it is essential for compatibility between the macro and the platform. The fact that Borland could theoretically have created a spreadsheet program incompatible with the 1-2-3 macros (and ultimately did so in 1992) is irrelevant. The critical point is that within the 1-2-3 environment, users and would-be competitors have no alternative to the 1-2-3 command structure.

***23 C. Judge Boudin Correctly Identified the Adverse Consequences of Extending Copyright Protection to the 1-2-3 Command Structure.**

In his concurring opinion, Judge Boudin observed that “[i]f Lotus is granted a monopoly on this pattern, users who have ... devised their own macros are locked into Lotus ...” 49 F.3d at 821. Judge Boudin further observed that granting a monopoly over a standard would have the additional undesirable effect of blocking development of a later, superior product:

Apparently, for a period Lotus 1-2-3 has had such sway in the market that it has represented the de facto standard for electronic spreadsheet commands. So long as Lotus is the superior spreadsheet -- either in quality or in price -- there may be nothing wrong with this advantage.

But if a better spreadsheet comes along, it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an investment in learning made by the users and not by Lotus. Lotus has already reaped a substantial reward for being first; assuming that the Borland program is now better, good reasons exist for freeing it to attract old Lotus customers: to enable the old customers to take advantage of a new advance

Id. at 821.

ACIS and CCIA believe that Judge Boudin's reasoning applies to all interface specifications. The extension of copyright protection to such interface specifications would be wrong as a matter of copyright law, and would effectively *24 eliminate competition in operating systems, or any software product that functions as a platform for other software products.

Such a broad monopoly would have serious implications for consumer welfare.¹³ In the absence of competition during the effective lifespan of the product, the first developer would have little incentive to develop more innovative and less costly products. These negative consequences would be compounded by the fact that the personal computer revolution, and recent major improvements in communications technology, have produced an overwhelming need for interconnection between different elements of computer systems. Within a given large corporation, literally thousands of personal computers and workstations scattered across the globe need to interact with each other and with the company's mainframes. Moreover, with the advent of the Global Information Infrastructure, different firms will need to exchange vast quantities of data through their computers.¹⁴ Copyright protection for interface specifications would lock users into a particular operating system or network software environment, and would inhibit the transfer of data between users with different computing environments.

It should be stressed that compatible products are not mere “clones” that offer only the same functionality as the products of the first comer, but at a lower price. While compatible products must offer at least the same functionality, they typically offer additional functionalities not found in the first comer's products. Borland's Quattro Pro is an *25 excellent example of this transformation. While Quattro Pro included a 1-2-3 mode to enable compatibility with 1-2-3 macros, it also offered users a native Borland mode completely different from 1-2-3. Moreover, even in the 1-2-3 mode, Quattro Pro inserted additional commands not present in the 1-2-3 command structure.

The compatible products developed by ACIS and CCIA members all add value to consumers in a similar manner. They compete with the first comers' products not only in terms of price (indeed, sometimes the compatible products may be more expensive), but also in terms of innovation. In this respect, compatible developers' use of preexisting interface specifications is a transformative use of the sort accredited by this Court in *Campbell v. Acuff-Rose Music, Inc.*, 114 S. Ct. 1164 (1994).

Some ACIS and CCIA companies are small start-ups with a handful of employees. Others are Fortune 500 companies. All need to interoperate with the computer systems developed by other firms in order to reach the market for their new products, which push the technological envelope. Reversal of the First Circuit's decision threatens to wipe out this vital sector of the U.S. software industry that has helped maintain our leadership position in global markets.

D. Lotus and Its Amici Grossly Exaggerate the Adverse Consequences, If Any, of the First Circuit's Decision.

Lotus states that “[t]he First Circuit's decision effectively nullifies the copyright protection Congress enacted for the expression in computer programs.” Lotus Brief at 42. Lotus reaches this breathtaking conclusion by pursuing the following reasoning: “If, as the First Circuit held, anything that can be defined as part of a method of operation is ineligible for copyright protection, then by the same logic virtually everything *26 in a computer program -- source code no less than textual menus on the screen -- is unprotected.” Id. at 18. As discussed above, the critical flaw in Lotus' reasoning is that the First Circuit did not hold that anything that can be defined as part of a method of operation is ineligible for copyright protection; rather, it held that anything that is an essential part of a method of operation is ineligible for copyright protection. The word “essential” makes all the difference in the world; it signals merger of idea and expression.

The First Circuit found that the selection and arrangement of the individual terms and menus had merged with the 1-2-3 command structure. The analysis then shifted to whether the command structure, taken as a whole, received copyright protection. The court concluded that it did not, in large measure because of its macro compatibility functionality.

Contrary to Lotus' assertion, the First Circuit's decision leaves many original program elements protected. A program's actual lines of code, and the structure of the code, remain protected to the extent they are not essential to achieving specific functionality. Here, Borland could achieve 1-2-3 compatibility using completely different code. Similarly, many features of a program's screen displays remain protected. Indeed, one can even imagine menu commands that might receive protection. If a menu had fanciful icons instead of words, for example, the individual icons might receive protection.

The Information Technology Industry Council, in its brief amicus curiae in support of Lotus' petition for a writ of certiorari, argued that the First Circuit's ruling “contains no limiting principle by which to differentiate the aspects of a program that are copyrightable from those that are not protectable.” ITIC Brief at 11. In fact, the First Circuit's decision *27 does contain such a limiting principle: those elements essential for programs to achieve the same functionality do not receive copyright protection. The First Circuit further holds that because compatibility is a functional characteristic, those elements necessary to achieve compatibility do not receive copyright protection. If these limiting principles are not clear enough from the face of the First Circuit's decision, then this Court should announce them unambiguously.

Lotus doubtlessly will complain that even a ruling restricted to compatibility elements will diminish software firms' incentive to innovate. To be sure, this ruling will increase the competition faced by first comers. But this competition will merely prevent first comers from appropriating the returns on the users' substantial investment in developing their own programs and files based on the first comers' products.¹⁵

Further, economists have demonstrated that in markets with strong network externalities, such as the market for software, the first comer reaps enormous competitive advantages from the establishment of a de facto standard interface specification.¹⁶ Copyright should not compound this *28 “winner takes all” result by locking the gateway to competition.

Lotus argues that antitrust law, rather than copyright law, should address the first comer's anticompetitive behavior. Lotus Brief at 47. A copyright, however, is a legal monopoly created by the Copyright Act, and the appropriate scope of that monopoly should be determined in the first instance by reference to the checks and balances contained in the Copyright Act itself, rather than the Sherman Act.

In any event, Lotus will still have ample incentive to develop improved versions of 1-2-3 as well as new products. As noted above, copyright will protect its implementation of programs that conform to the 1-2-3 command structure or other interface specifications that it develops in the future.

Moreover, Lotus can seek patent protection for its command structures. In *Diamond v. Diehr*, 450 U.S. 175 (1981), this Court opened the gates to software patents. Recent Federal Circuit decisions have pushed the gates open even further. E.g., In *Alappat*, 33 F.3d 1526 (Fed. Cir. 1994).¹⁷ Under the evolving case law, many aspects of computer interfaces, including many command structures such as the one at issue here, may be patentable subject matter. Indeed, even before the above legal developments, companies were able to obtain utility patents on menu command hierarchies. J.A. 842-46, 856-63 (portions of U.S. Patent Nos. 4,989,141 and 4,611,306).

***29** If interface specifications were protected under the copyright laws, the public would not have the same safeguards against overbroad monopolies that exist under the Patent Act. The sole requirement for a copyright is that it be an original, expressive work of authorship. 17 U.S.C. § 102(a). By contrast, an invention must satisfy the novelty requirements of 35 U.S.C. § 102 and the non-obviousness requirements set forth in 35 U.S.C. § 103. Furthermore, the specification must contain a sufficiently detailed description of the invention to enable one skilled in the relevant art to practice it, and must disclose the best mode contemplated by the inventor for carrying out the invention. 35 U.S.C. § 112. If a patent applicant must satisfy these standards to obtain a 20-year monopoly on a method of operating a computer, it is difficult to understand why that same inventor should obtain copyright protection in the same subject matter for 75 years or longer, without even satisfying those standards.

ACIS and CCIA members rely on copyright protection for many aspects of computer programs, and consider it sufficient for its intended purposes. ACIS and CCIA members also frequently obtain patents when warranted. This includes hardware and software processes such as the “selection and arrangement of executable operations.” ACIS and CCIA are deeply concerned, however, that a software company might obtain patent-like protection for functional characteristics of software that do not meet the rigorous standards of patentability, merely by claiming a copyright. The First Circuit's refusal to allow Lotus to employ copyright law to obtain a de facto patent is consistent with this Court's refusal to allow Bonito Boats to employ Florida state law to obtain a de facto patent.

CONCLUSION

In order to achieve compatibility with macros written by the third-party users, it was necessary for Borland to use the ***30** words of the 1-2-3 command structure, and to display those words on the computer screen. In using these words, Borland reproduced only the minimum portions of Lotus 1-2-3 necessary to achieve macro-compatibility, and nothing more. Creators of computer programs such as Borland should not be prevented by copyright law from using as much of another program's utilitarian methods of operation as is necessary to enable their programs to work with other programs. This is needed to ensure interoperability across computer systems and networks, and interoperability is needed to promote consumer welfare on the terms contemplated by the Copyright Act. The First Circuit correctly recognized the force of these principles in the applicable law. Under the First Circuit's holding, copyright will provide a secure basis for continued innovation and competition in the software industry.

For the above reasons, the decision of the First Circuit should be affirmed. This Court should hold that because compatibility is a functional characteristic, those program elements necessary to achieve compatibility do not receive copyright protection.

Footnotes

1 ACIS members include Accolade, Inc., Advanced Micro Devices, Amdahl Corporation, America Online, Inc., AT&T Global Information Solutions, Broderbund Software, Inc., Bull HN Information Systems, Inc., Chips and Technologies, Inc., Clearpoint Research Corporation, Color Dreams, Inc., Comdisco, Inc., Emulex Corporation, Forecross Corporation, The Fortel Group, Fujitsu

Systems Business of America, Inc., Hitachi Data Systems, ICTV, Insignia Solutions, Integrated Documents Applications Corp., Johnson-Laird, Inc., Landmark Systems Corporation, LCS/Telegraphics, MidCore Software, Inc., New York Systems Exchange, Inc., Octel Communications Corporation, Phoenix Technologies, Ltd., Plimoth Research Inc., QAD Inc., Seagate Technology, Inc., Software Association of Oregon, Storage Technology Corporation, Sun Microsystems, Inc., Tandem Computers, Inc., 3Com Corporation, Trilium Consumer Electronics, Inc., Western Digital Corporation, and Zenith Data Systems Corporation. (The Software Association of Oregon consists of over 550 software development firms, firms in associated industries, and individuals professionally involved in software development.)

2 CCIA members include Amdahl, AT&T, Bell Atlantic, Cambridge Technology Partners, Charles River Data Systems, Datum, Inc., Formation, Fujitsu Network Switching, Hitachi Data Systems, InterCAP Graphic Systems, Leasing Solutions, Northern Telecom, NYNEX, Okidata, Storage Technology Corporation, Summa Four, Inc., Sun Microsystems, Inc., Tandem Computers, Inc., TSI International Software Ltd., and ViON Corporation.

3 IBM (Lotus' parent) and amicus Intel both have also used copyright law to protect and extend their market power. See J. Band and M. Katoh, *Interfaces on Trial* (1995) at 26-28, 35-39.

4 The American Intellectual Property Law Association ("AIPLA") Brief is hardly a brief "in support of neither party." AIPLA Brief at 1. It makes virtually the same arguments as the Lotus Brief and the joint Digital Equipment, Gates Rubber, Intel and Xerox brief ("DEC Brief"). This comes as no surprise, given that one of the law firms on the AIPLA brief represents a party that would benefit from a reversal of the First Circuit's decision in a case pending in the Eleventh Circuit, *Mitek Holdings, Inc. v. Arce Engineering Co.*, No. 94-5262 (11th Cir. filed Nov. 30, 1994) (appeal from 864 F. Supp. 1568 (S.D. Fla. 1994)).

5 U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (U.S. Government Printing Office, Washington, D.C., May 1992) ("OTA Report") at 26.

6 See, e.g., J. Sandberg and G. Hill, *Microsoft Probe Spurs Subpoenas Tied to Internet*, *Wall St. J.*, Dec. 4, 1995; J. Sandberg, *Sun and Netscape Are Forming Alliance Against Microsoft on Internet Standard*, *Wall St. J.*, Dec. 4, 1995.

7 See, e.g., 3 M. Nimmer & D. Nimmer, *Nimmer On Copyright*, § 13.03[F] (1995); *Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2nd Cir. 1992).

8 The Whelan court also held that copyright protects the expression in the non-literal elements of a computer program. This holding, a simple restatement of the fundamental copyright principle that copyright protects original expression in works of authorship, is non-controversial and has been readily accepted by other courts.

9 Prior to the Argument section of its Brief, Lotus appears to agree with this formulation of the issue: "It is the copyrightability of the overall combination of words and menus in the 1-2-3 menu command hierarchy, viewed as a whole, and not any individual menu command ... viewed in isolation, that is at issue in this case." Lotus Brief at 6 (footnote omitted).

10 In this context, it is worth noting that Microsoft's spreadsheet product Excel at one time also had a file like Borland's Key Reader which permitted 1-2-3 macro compatibility. Lotus did not challenge Microsoft in court, but instead permitted Microsoft's product to remain compatible with 1-2-3 while Borland was precluded by court injunction from offering a compatible product. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223, 229-30, 234 (D. Mass. 1993), rev'd, 49 F.3d 807 (1st Cir.), and cert. granted, 116 S. Ct. 39 (1995).

The reason Excel ultimately eclipsed Lotus 1-2-3 is a further demonstration of the significance of interface specifications and compatibility: Microsoft introduced a new operating system (Windows), migrated the installed base of computer users from the old operating system (DOS) to the new one (Windows), and released a Windows-compatible version of Excel before providing Lotus with sufficient interface information to permit the release of 1-2-3 for Windows. See C. Morris and C. Ferguson, *How Architecture Wins Technology Wars*, *Harv. Bus. Rev.* (Mar./Apr. 1993). Indeed, Lotus vociferously complained to the Antitrust Division of the Department of Justice about Microsoft's conduct in manipulating the forces of the free market to achieve a monopoly. See, e.g., M. Knell, *Microsoft Fights for Intuit*, *The Boston Herald*, Apr. 28, 1995, at 23.

11 See *Interfaces on Trial* at 87-90. In the trademark context, this Court has explained that "'a product feature is functional,' and cannot serve as a trademark, 'if it is essential to the use or purpose of the article or if it affects the cost or quality of the article,' that is, if exclusive use of the feature would put competitors at a significant non-reputation related disadvantage." *Qualitex Co. v. Jacobson Products Co.*, 115 S. Ct. 1300, 1304 (1995) (quoting *Inwood Laboratories, Inc. v. Ives Laboratories, Inc.*, 456 U.S. 844, 850 n.10 (1982)).

12 See also *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 46 F.3d 408, 410 (5th Cir. 1995), supplementing opinion, 26 F.3d 1335 (5th Cir. 1994) (copyright protection should not "extend to the manufacturing of computer hardware so as to deter achieving compatibility with other models"). The district court below stated that these appellate decisions simply stood for the proposition that

copyright allows a new comer software developer to replicate the design of the first comer's "plug" in order to plug his software into the first comer's "socket," but that copyright nonetheless prohibits the new comer from replicating the design of the "socket" in order to satisfy the demand for additional outlets. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 212-13 (D. Mass. 1992), rev'd, 49 F.3d 807 (1st Cir.), and cert. granted, 116 S. Ct. 39 (1995). The district court reached this metaphysical conclusion by reasoning that the socket was created before the programs with the compatible plugs. The precise order of creation, however, should make little difference; if copyright does not protect the design of the plug, it surely does not protect the design of the socket. Moreover, even if the socket was actually constructed before the plugs, the socket and the plug obviously were designed at the same time so as to fit into one another.

- 13 See, e.g., P. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 *Stan. L. Rev.* 1045, 1082, 1097 n.281 (1989).
- 14 See President's Information Infrastructure Task Force, *Global Information Infrastructure: Agenda for Cooperation* (U.S. Government Printing Office, Washington, D.C., Feb. 1995) at 14-16.
- 15 See F. Warren-Boulton, K. Baseman & G. Woroch, *Copyright Protection of Software Can Make Economic Sense*, 12 *Computer Law* 10, 23 (Feb. 1995)
- 16 See G. Saloner, *Economic Issues in Computer Interface Standardization*, 1 *Econ. Innov. New. Tech.* 135, 140 (1990):
Because of the compatibility and network benefits, all else equal, a new user prefers a vendor with a larger total installed base of users. Thus installed bases have a tendency to be self-perpetuating: they provide the incentive for the provision of products (software and hardware) that is compatible with that in the installed based which in turn attracts new users to the installed base further swelling its ranks and increasing the incentive for the provision of even more complementary products.
See also *United States v. Microsoft Corp.*, 56 F.3d 1448, 1452 (D.C. Cir. 1995); R. Frank and P. Cook, *The Winner-Take-All Society* (1995).
- 17 See also U.S. Patent and Trademark Office, *Proposed Examination Guidelines for Computer-Implemented Inventions*, 60 *Fed. Reg.* 28,778 (1995).