1993

Assaying *Computer Associates v. Altai*: How Will the Golden Nugget Test Pan Out

Walter Effross

ASSAYING *COMPUTER ASSOCIATES v. ALTAI*: HOW WILL THE "GOLDEN NUGGET" TEST PAN OUT?

WALTER A. EFFROSS*

* Associate, McCarter & English, Newark, New Jersey. B.A. 1984 *cum laude*, Princeton University; J.D. 1987, Harvard Law School. The author appreciates the comments of Charles R. Merrill and Myrna L. Wigod of the Computer and High Technology Group of McCarter & English.

> Once a court has sifted out all elements of the allegedly infringed program which are "ideas" or are dictated by efficiency or external factors, or taken from the public domain, there may remain a core of protectable expression. In terms of a work's copyright value, this is the golden nugget.
>
> — Hon. John M. Walker, Jr.[1]

> All, as they say, that glitters is not gold.
>
> — John Dryden[2]

I. INTRODUCTION

Over the past decade, courts have struggled to apply to computer software the copyright standards developed since the late eighteenth century[3] to protect books and plays. Those traditional "literary works," which consist of fixed sequences of words, directly reveal all levels of their structure, from individual phrases to broad themes and plot elements, to their readers and viewers. By contrast, computer programs generally consist of coded instructions that would be unintelligible to the average computer user, who usually interacts with them only through the simplifying medium of the program's "user interface." The relative "opaqueness" of a program's structure to the layman reflects his concern with its practical application over its architectural aesthetics: to use a spreadsheet, word processor, or video game, it

---

1. Computer Assocs. Int'l, Inc. v. Altai, Inc., Nos. 91-7893, 91-7935, 1992 U.S. App. LEXIS 14305 (2d Cir. June 22, 1992), *modified*, 982 F.2d 693, 710 (2d Cir. 1992).

2. THE HIND AND THE PANTHER, part 2, line 215 *in* THE POEMS AND FABLES OF JOHN DRYDEN (James Kinsley, ed., Oxford University Press 1962) (1687). *Cf.* WILLIAM SHAKESPEARE, MERCHANT OF VENICE act 2, sc. 7 ("All that glisters is not gold.") (Stanley Wells & Gary Taylor eds., Oxford University Press 1986) (1596/97); EDMUND SPENSER, FAERIE QUEENE, bk. 2, canto 7, stanza 14 ("Gold is not all that doth golden seem.") (Thomas P. Roche, ed., Yale University Press 1981) (1590); GEOFFREY CHAUCER, THE CHANOUN YEMANNES TALE, line 962 ("But al thing which that shyneth as the gold/Nis nat gold, as that I have herd it told.") *in* THE CANTERBURY TALES (Walter Skeat, ed., Oxford University Press 1972) (c. 1390).

3. OFFICE OF TECHNOLOGY ASSESSMENT, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY AND THE CHALLENGE OF TECHNOLOGICAL CHANGE 59 (Washington, D.C.: U.S. Gov't Printing Office 1992) (copyright statutes enacted by American colonies in 1780s) [hereinafter FINDING A BALANCE].

should not be necessary for him to "look under the hood" of the software.

Like the sentences of books and plays, the exact written instructions by which a program instructs the computer to process data are protected by copyright. More controversial, however, has been the copyright status of the "non-literal elements" of computer programs, that is, of software's "structure, sequence and organization," which correspond to the "plot, subplot, sequence of scenes, setting, characterization and patterns of dialogue in works of fiction or dramas; or . . . the detailed outline and organization and selection, coordination and arrangement of information in textbooks or other nonfiction works."[4]

To determine which elements of a computer program may be copyrighted, and when such a copyright has been infringed, courts have revisited the first principles of intellectual property law, such as the distinction between (uncopyrightable) ideas and (copyrightable) expressions, and the test for "substantial similarity" between original and allegedly infringing works. In *Computer Associates International, Inc. v. Altai, Inc.* the Second Circuit incorporated a variety of standard copyright principles into a three-part "abstraction/filtration/comparison" analysis. Under this procedure, a court analyzes a program from its most specific to most general levels, strips away uncopyrightable material to isolate a "golden nugget" of protectable (and connectable) elements, and determines whether there has been unauthorized copying of this copyrightable core. Though this test efficiently combines familiar approaches to the protection of intellectual property, it nonetheless raises both theoretical and practical questions of its own.

Part II of this Article discusses the evolution of copyright protection for literal and non-literal elements of software. The factual background and legal analysis of the *Computer Associates* decision are reviewed in Part III. Part IV examines the implications of the Second Circuit's new standard for copyrightability, from its effect on competition in the software market to its impact on the use of expert testimony and on the relative merits of

4. *Id.* at 69.

copyrights and patents. Finally, Part V addresses the initial judicial response to the Second Circuit's test, including its incorporation into two Court of Appeals decisions upholding reverse engineering as a "fair use" of software.[5]

II. BACKGROUND

A. Protection of Literal Elements of Software

Because of their textual nature, the instructions that constitute a computer program were accorded copyright protection relatively quickly: a special Congressional commission expanded the statutory category of "literary works" eligible for such treatment.

Article I, Section 8, Clause 8 of the United States Constitution authorizes Congress "[t]o promote the Progress of Science and Useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." The Copyright Act of 1976[6] extended copyright protection to "original works of authorship fixed in any tangible medium of expression . . . from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."[7] As one of the seven categories

5. The decisions analyzed in Part V refer to the *Computer Associates* opinion as it was originally released on June 22, 1992. *See supra* note 1. This opinion was withdrawn by the Second Circuit and superseded on rehearing by an opinion handed down on December 17, 1992. The revised opinion did not amend the enunciation or discussion of the "abstraction-filtration-comparison" test for copyright infringement of software, on which this Article focuses. Rather, the Second Circuit vacated, where it had earlier affirmed, the district court's ruling that Computer Associates' trade secret claims had been preempted. *See infra* note 186. The *Computer Associates* decision was released on June 22, 1992. This Article covers developments in the area through January 31, 1993.

6. Act of October 19, 1976, Pub. L. No. 94-553, 90 Stat. 2541 (codified at 17 U.S.C. §§ 101-810 (1988)) [hereinafter the Act]. This Act superseded the Copyright Act of 1909.

7. 17 U.S.C. § 102(a) (Supp. II 1990). Works are "fixed in [a] tangible medium of expression" when their:

> embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration. A work consisting of sounds, images, or both, that are being transmitted, is "fixed" for the purposes of this

of "works of authorship," "literary works" includes "works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied."[8]

A House Report accompanying the Act indicated that the issues relating to the copyright status of computer programs were "not sufficiently developed for a definitive legislative solution,"[9] but stated that software fell within the category of "literary works."[10]

In 1974, Congress created a Commission on New Technological Uses (CONTU) to scrutinize this issue, among others.[11] CONTU's Final Report, issued in 1978,[12] proposed that Section 101 of the Act define "computer programs"[13] and that the copyright law "make it explicit that computer programs, to the extent that they embody an author's original creation, are proper subject matter of copyright."[14] Accordingly, the 1980 amendments[15] to the Act[16] brought within the scope of copyright

---

title if a fixation of the work is being made simultaneously with its transmission.

Id. § 101.

8. *Id.* § 101.

9. H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54, 116 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5731. However, § 117 of the Act addressed computer uses of copyrighted works. 17 U.S.C. § 117.

10. The Report defined "literary works" to include "computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves." H.R. REP. NO. 1476, *supra* note 9, at 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667.

11. Created by Pub. L. No. 93-573, § 201, 88 Stat. 1873 (1974).

12. NATIONAL COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT (1979) [hereinafter CONTU REPORT].

13. *Id.* at 12.

14. *Id.* at 1.

15. H.R. REP. NO. 1307, 96th Cong., 2d Sess. 23 (1980), *reprinted in* 1980 U.S.C.C.A.N. 6460, 6482. *See* Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 750 n.6 (N.D. Ill. 1983) ("Although the Congressional action in 1980 does not appear to be supported by a legislative history, it is fair to conclude, since Congress adopted its recommendations without alteration, that the CONTU Report reflects the Congressional intent.").

16. Act of Dec. 12, 1980, Pub. L. No. 96-517, § 10, 94 Stat. 3015, 3028.

protection a "computer program," defined 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]

The "first wave"[18] of software copyright cases involved the relative protection to be granted to a program's "source code," that is, to instructions "written in any of several programming languages employed by computer programmers."[19] The amendment's phrase, "directly or indirectly," along with the Act's existing policy of extending copyright protection to "original works of authorship . . . [that] can be perceived . . . with the aid of a machine or device,"[20] led courts to expand the scope of such copyright protection to include not only human-readable "source code" but also machine-readable "object code,"[21] the form in which the program's instructions are implemented by the computer.[22]

---

17. 17 U.S.C. § 101 (emphasis added).

18. David Bender, *Computer Associates v. Altai: Rationality Prevails*, COMPUTER LAW., Aug. 1992, at 1, 2.

19. CONTU REPORT, *supra* note 12, at 21 n.109.

20. 17 U.S.C. § 102(a).

21. Source code as written by the programmer is transformed "within the computer, through intervention of a so-called compiler or assembler program, into an 'object code.' This last is most often physically embodied in the present state of technology, in punched cards, magnetic disks, magnetic tape, or silicon chips." CONTU REPORT, *supra* note 12, at 28.

As distinguished from object code, which refers to statements written in "machine language, a binary language using two symbols, 0 and 1, to indicate an open or closed switch," source code can be either "high level language, such as the commonly used BASIC or FORTRAN, [which] uses English words and symbols, and is relatively easy to learn and understand" or the interim form of "assembly language, which consists of alphanumeric characters." Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) (protecting both object code and source code of operating system program).

22. *See, e.g.*, Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 750 (N.D. Ill. 1983) ("It is certain as a general matter that the current copyright legislation is intended to protect object code as well as source code."); Hubco Data Prods. Corp. v. Management Assistance, Inc., 219 U.S.P.Q. 450, 454 (D. Idaho 1983) (since CONTU Report and Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982), specifically recognize copyrightability of object code under 17 U.S.C. § 102, plaintiff has a reasonable probability of success on such claim); GCA Corp. v. Chance, 217 U.S.P.Q. 718, 720 (N.D. Cal. 1982) ("Because the object code is the encryption of the copyrighted source code, the two are to be treated as one work; therefore, copyright of the source code protects the object

Thus, the Third Circuit, noting that "the copyrightability of computer programs is firmly established after the 1980 Amendments to the Copyright Act," rejected the notion that copyright protection applied only to work "intended as a medium of communication to human beings,"[23] which qualification would have excluded object code. Similarly, computer instructions embedded electronically in the "ROM" computer chips of a circuit board[24] were accorded the same protection as those instructions written on paper by programmers.[25] One year later, the court reiterated that "a computer program, whether in object code or source code, is a 'literary work' and is protected from unauthor-

---

code as well."); Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171 (N.D. Cal. 1981).

23. Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 877 (3d Cir. 1982) (finding copyright infringement of object code responsible for the audiovisual aspects of video game).

24. The electronic circuitry [of the coin-operated video game at issue in *Williams Electronics*] includes a microprocessor and memory devices, called ROMs (*R* ead *O* nly *M* emory), which are tiny computer 'chips' containing thousands of data locations which store the instructions and data of a computer program. The microprocessor executes the computer program to cause the game to operate.

Id. at 872.

25. *Id.* at 874. The Second Circuit has also upheld the copyrightability of program elements "imprinted" on PROM's, "Programmable Read Only Memory" chips. "The stored information in a ROM cannot be changed; it is imprinted into the ROM when the device is manufactured. A PROM is a ROM into which information can be imprinted (programmed) after manufacture; once the information is programmed in a PROM, it cannot be changed simply by writing in a new program." Stern Elecs. v. Kaufman, 669 F.2d 852, 855 n.1 (2d Cir. 1982) (source code, as "the written computer program," eligible for copyright; whether located in PROM prepared for video game in question or elsewhere in game apparatus, all portions of program, once stored in memory devices anywhere in game, are "fixed in a tangible medium" for purposes of copyrightability). *Cf.* Cable/Home Comm. Corp. v. Network Prods., Inc., 902 F.2d 829 (11th Cir. 1990) (concerning copyright infringement of computer programs fixed in silicon chips contained in integrated circuit of video signal descrambler).

But see Commissioner Hersey's dissent from the CONTU Report, to the effect that "copyright protection does not extend to a computer program in the form in which it is capable of being used to control computer operations." CONTU REPORT, *supra* note 12, at 1. Commissioner Hersey distinguished between traditional "[w]orks of authorship [which] have always been intended to be circulated to human beings and to be used by them" and "[c]omputer programs, [which] in their mature phase, are addressed to machines." *Id.* at 28.

ized copying, whether from its object or source code version."[26]

By 1989, the Ninth Circuit could assert that "[s]ource and object code, the literal components of a program, are consistently held protected by a copyright on the program."[27] However, the more problematic protection of the "non-literal components of a program, including [its] structure, sequence and organization and user interface," soon gave rise to a "second wave" of software copyrightability cases.[28] The determination of copyrightability of non-literal elements was not only extremely fact-sensitive[29] but was also subject to two seemingly flexible standards: the distinction between "idea" and "expression" and the evaluation of two programs' "substantial similarity."[30] These standards determine,

---

26. Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983) (protecting object code version of operating system). *See also* Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521 (9th Cir. 1984) (object code is copyrightable).

27. Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173, 1175 (9th Cir. 1989) (concluding that nonliteral components of computer software may be protected by copyright where they constitute expression rather than ideas) (citing CMS Software Design Sys., Inc. v. Info Designs, Inc., 785 F.2d 1246, 1249 (5th Cir. 1986) (suggesting that source code is copyrightable) and Apple Computer Inc. v. Formula Int'l, Inc., 725 F.2d 521 (9th Cir. 1984) (object code is copyrightable)).

See also Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 45 (D. Mass. 1990) (parties agree as a general proposition that source code and object code, if original, are copyrightable); Digital Communications Assocs., Inc. v. Softklone Distrib. Corp., 659 F. Supp. 449, 454 (N.D. Ga. 1987) ("Case law under the [Copyright] Act also clearly establishes that copyright protection extends both to a program's source code, written in conventional human language and symbols, and object code, written in machine readable binary language.") (citing Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984)).

28. Bender, *supra* note 18, at 2.

29. *Johnson Controls*, 886 F.2d at 1175. The Ninth Circuit found that the relevant inquiry would concern whether "the component in question qualifies as an expression of an idea, or an idea itself." *Id.*

30. "[T]he determination of the extent of similarity which will constitute a *substantial* and hence infringing similarity presents one of the most difficult questions in copyright law, and one which is the least susceptible to helpful generalizations." 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[A], at 13-27 (Matthew Bender ed., 1992) (emphasis in original).

See also Anthony L. Clapes et al., *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 U.C.L.A. L. REV. 1493, 1571 (1987) ("A determination of substantial similarity

respectively, the degree to which copyright protection will be extended to a computer program and the manner in which an allegation of copyright infringement of that program will be investigated.

B. *Protection of Non-Literal Aspects of Software*

Although the source code and object code of software may comfortably be compared to the text of books and plays, the functional nature of computer programs limits the extension of the analogy beyond the "literal" structures of these works. Where different sets of coded instructions are alleged to have produced programs of similar arrangement and operation, courts have reexamined the venerable distinction between the uncopyrightable "ideas" underlying a work and copyrightable "expressions" of those ideas in the work itself. Once the copyrightable elements of a program have been identified, their unlawful duplication by another program has been determined by a two-part test involving "substantial similarities" between the programs.

1. *Idea/Expression Distinction*

A fundamental principle of copyright law is that such protection applies only to expressions of ideas, and not to the ideas themselves. In the seminal case of *Baker v. Selden*,[31] the Supreme Court characterized as an uncopyrightable idea the plaintiff's innovative method of accounting, and found the printed forms produced by the plaintiff for the implementation of his system were uncopyrightable, "as necessary incidents to the art"[32] of the new method. The Court thus denied the plaintiff recourse against an alleged infringer of these forms.

The Copyright Act reflects the idea/expression distinction, by providing that "[i]n no case does copyright protection for an

---

in a particular [software] case implicates, to a certain degree, a species of the famous 'I know it when I see it' test. This is because authorship, virtually by definition, is highly individualistic and the means of plagiarizing an author's expression cannot be precast into inevitable, predefined categories. . . .").

31. 101 U.S. 99 (1879).

32. *Id.* at 103.

original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."[33]

Nonetheless, the line dividing idea from expression is often elusive. Judge Learned Hand recognized that

> [t]he test for infringement of a copyright is of necessity vague. In the case of verbal "works" it is well settled that although the "proprietor's" monopoly extends beyond an exact reproduction of the words, there can be no copyright in the "ideas" disclosed but only in their "expression." Obviously, no principle can be stated as to when an imitator has gone beyond copying the "idea," and has borrowed its "expression." Decisions must therefore inevitably be ad hoc.[34]

The distinction between "idea" and "expression" has proven particularly troublesome in the context of software. Although CONTU addressed this issue,[35] courts have grappled with the "hybrid nature of a computer program, which, while it is literary

---

33. 17 U.S.C. § 102(b). The legislative history reveals § 102 was intended "to make clear that this 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 the copyright law." H.R. REP. NO. 1476, 94th Cong., 2d Sess. 57 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670, *quoted in* Allen-Myland, Inc. v. IBM Corp., 746 F. Supp. 520, 532 (E.D. Pa. 1990). *See also* Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1234 (3d Cir. 1986), *cert. denied,* 479 U.S. 1031 (1987); *cf.* 35 U.S.C. § 101 (Patent Act protects, *inter alia*, processes and machines).

34. Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487, 489 (2d Cir. 1960) (protecting *pendente lite* ornamental design printed on cloth, in light of high degree of resemblance between patterns).

35. The CONTU Report indicates that:

> The "idea-expression identity" exception provides that the copyrighted language may be copied without infringing when there is but a limited number of ways to express a given idea. This rule is the logical extension of the fundamental principle that copyright cannot protect ideas. In the computer context this means that when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement . . . .
>
> When other language *is* available, programmers are free to read copyrighted programs and use the ideas embodied in them in preparing their own works.

CONTU REPORT, *supra* note 12, at 20, *quoted in* Allen-Myland, Inc. v. IBM

expression, is also a highly functional, utilitarian component in the larger process of computing."[36]

2. *Substantial Similarity*

To demonstrate infringement, a copyright owner must prove two elements: (1) that it owns a valid copyright and (2) that the defendant copied copyrightable portions of the plaintiff's work.[37]

Although the second of these elements can be difficult to demonstrate directly,[38] such proof is not required by the Second Circuit's standard test for copyright infringement, enunciated in *Arnstein v. Porter*.[39] Each phase of the *Arnstein* inquiry involves its own "substantial similarity" analysis. First, copying can be inferred from the alleged infringer's access to the plaintiff's work if the works exhibit a "substantial similarity" that cannot be attributed to the defendant's independent duplication of the plaintiff's efforts.[40] Indeed, even if the plaintiff cannot demonstrate access to its work, copying will be established if the similarities between the works are "so striking as to preclude the possibility of" independent development.[41] Expert testimony is admissible

Corp., 746 F. Supp. 520, 532 n.8 (E.D. Pa. 1990) (CONTU Report serves as legislative history of provisions recommended by CONTU).

36. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 712 (2d Cir. 1992). *See also* Steven R. Englund, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 893 (1990) (difficulty with relying upon analogies to literary works is that computer programs are primarily utilitarian in nature).

37. Novelty Textile Mills, Inc. v. Joan Fabrics Corp., 558 F.2d 1090, 1092 (2d Cir. 1977). *See also* Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204, 206 (9th Cir. 1988); Atari, Inc. v. North Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 614 (7th Cir.), *cert. denied*, 495 U.S. 880 (1982); Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp., 562 F.2d 1157, 1162 (9th Cir. 1977); Reyher v. Children's Television Workshop, 533 F.2d 87, 90 (2d Cir.), *cert. denied*, 429 U.S. 980 (1976); 3 NIMMER & NIMMER, *supra* note 30, § 13.01.

38. Roth Greeting Cards v. United Card Co., 429 F.2d 1106, 1110 (9th Cir. 1970) (direct evidence of copying is rarely available).

39. 154 F.2d 464, 468 (2d Cir. 1946) (comparison of Cole Porter's musical compositions with those of plaintiff disclosed similarities sufficient to support finding of infringement if Porter could have had access to plaintiff's works).

40. "Of course, if there are no similarities, no amount of evidence of access will suffice to prove copying." *Id.*

41. *Id.*

in conducting an exacting comparison, or "dissection," of the works to identify the elements of similarity.[42]

Second, once copying has been demonstrated, the plaintiff must also show that the substantial similarities between the works arise from the duplication of the copyrighted elements of its own work.[43] This proof involves the "ordinary lay [opinion]" and not expert testimony.[44]

In *Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp.*,[45] the Ninth Circuit dubbed *Arnstein*'s two "substantial similarity" tests the "extrinsic test," for the duplication of uncopyrightable ideas, and the "intrinsic test," for the duplication of copyrightable expressions.[46] The "extrinsic test," which "depends not on the responses of the trier of fact, but on specific criteria which can be listed and analyzed" by the process of "ana-

---

42. *Id.*

43. Computer Associates Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 557 (E.D.N.Y. 1991) (quoting Walker v. Time Life Films, Inc., 784 F.2d 44, 48 (2d Cir.), *cert. denied,* 476 U.S. 1159 (1986) (in turn quoting Hoehling v. Universal City Studios, Inc., 618 F.2d 972, 977 (2d Cir.), *cert. denied,* 449 U.S. 841 (1980))), *aff'd in part, vacated in part,* 982 F.2d 693 (2d Cir. 1992). This sequence of elements was first enunciated by *Arnstein v. Porter* where the inquiry involved "whether defendant took from plaintiff's works so much of what is pleasing to the ears of *lay listeners,* who comprise the audience for whom such popular music was composed, that defendant wrongfully appropriated something which belongs to the plaintiff." *Arnstein,* 154 F.2d at 469, 473 (emphasis added).

44. As Arnstein v. Porter involved "an issue of fact which a jury is peculiarly fitted to determine . . . even if there were to be a trial before a judge, it would be desirable (although not necessary) for him to summon an advisory jury on this question." 154 F.2d at 469. Indeed, the Second Circuit itself, in rejecting the lower court's award of summary judgment in favor of the defendant, did not compare the sheet music of the works, as a musical expert might have, but instead listened to the compositions as played in the phonograph recordings submitted by the defendant. *Id.*

45. 562 F.2d 1157, 1164 (9th Cir. 1977) (finding copyright infringement of plaintiff's "H. R. Pufnstuf" television show by defendant's television commercials).

46. *Cf.* Concrete Mach. Co. v. Classic Lawn Ornaments, 843 F.2d 600, 606 (1st Cir. 1988) (under *Arnstein* test, trivial or minor variation between concrete representations of animals will not preclude finding of infringement). "Substantial similarity is an elusive concept, not subject to precise definition. It refers only to the expression of the artist's concept, not the underlying idea itself; mere identity of ideas expressed by two works is not *substantial* similarity giving rise to an infringement action." *Id.* (emphasis added).

lytic dissection" and by experts,[47] evaluates the degree to which uncopyrightable ideas have been duplicated.

The second *Arnstein* prong, or the "intrinsic test," is applied if the extrinsic test reveals "substantial similarity in ideas, [since] then the trier of fact must decide whether there is substantial similarity in the expressions of ideas so as to constitute infringement."[48] This test, which involves neither expert opinion nor analytic dissection, "does not depend on the type of external criteria and analysis which marks the extrinsic test."[49] Rather, the "response of the ordinary reasonable person" determines whether copying impermissibly extended to the protected "expression" of an idea.[50]

In holding that the "McDonaldland" television commercials infringed the copyrights of the "Living Island" scenario featured on the children's television series, "H. R. Pufnstuf," the Ninth Circuit emphasized that the inquiry of the intrinsic test extended beyond the analytic dissection's mere comparison of elements to encompass more comprehensive aspects of a work. Although there were distinctions between the characters, setting and plot of the two sets of television presentations, the extrinsic test's focus on these alone would ignore McDonalds' duplication of the underlying "intrinsic quality" which had made the "H. R. Pufnstuff" series the most popular Saturday morning programming for children.[51] In other words, "[w]e do not believe that the ordinary reasonable person, let alone a child, viewing these works will even notice that Pufnstuf is wearing a cummerbund while Mayor McCheese is wearing a diplomat's sash."[52]

---

47. Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp., 562 F.2d 1157, 1164 (9th Cir. 1977). "Such criteria include the type of artwork involved, the materials used, the subject matter, and the setting for the subject." *Id.*

48. *Id.*

49. *Id.*

50. *Id.*

51. *Id.* at 1166.

52. *Id.* at 1167. *Krofft*'s extrinsic test has been attacked as adding little to the determination of copyright infringement, since "it is not at all clear how one could have substantial similarity of expressions without having substantial similarity of ideas" and for "implicitly assum[ing] that one has already determined the idea/expression line." William E. Hilton, *Quantifying Originality: A Logi-*

Though the intrinsic test may penetrate to similarities between the broadest structures of two works, it can still be informed by analytic dissection's feature-by-feature comparison (and thus, by expert opinion). The Ninth Circuit's recent decision in *Brown Bag Software v. Symantec Corp.*[53] finds analytic dissection relevant to the intrinsic test's identification of copyrighted features that have been copied into the defendant's program. "Under the reformulated extrinsic test, we mean to perpetuate 'analytic dissection' as a tool for comparing not only ideas but also expression."[54]

The district court, after comparing in detail the "user interfaces"[55] of the programs at issue, had denied summary judgment to the defendant.[56] In affirming this part of the holding, the Ninth Circuit observed that *Krofft* does not prohibit comparing program elements to determine whether copyrightable elements of the plaintiff's program were copied by the defendant: in this context, "analytic dissection is used not for the purposes of comparing similarities and identifying infringement, but for the pur-

---

cal Analysis for Determining Substantial Similarity in Computer Software Copyright Infringement Actions, 31 IDEA: J.L. & TECH. 269, 289 (1991).

53. 960 F.2d 1465 (9th Cir.), *cert. denied,* 113 S. Ct. 198 (1992).

54. *Id.* at 1475.

55. "Generally, this interface relates either to the display screens projected on the cathode ray tube by the program (permitting the user to select various options and/or to input data in prescribed format), or to the use of specific keys, on the various standard keyboards, to perform particular functions." Bender, *supra* note 18, at 2.

56. Telemarketing Resources v. Symantec Corp., 12 U.S.P.Q.2d (BNA) 1991 (N.D. Cal. 1989), *aff'd sub nom.* Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir.), *cert. denied sub. nom.* BB Asset Management, Inc. v. Symantec Corp., 113 S. Ct. 198 (1992).

Brown Bag had contended that both outlining programs:

> (1) Begin with a menu labeled "OPENING MENU," which is shaded in a different color than the screen background and surrounded by a single line box;
>
> (2) Allow the user to select four of the same initial options, either by means of a "highlighting bar" (through use of the cursor arrows and "enter" key) or by typing the capital letters corresponding to the options;
>
> (3) Display, in similar color schemes, a help line on the bottom of the editing screen; a line around the screen; and, across the top of the

pose of defining the scope of plaintiff's copyright."[57] For example, four years previously, the court had dismissed the similarities between two video games depicting karate matches, as inherent in the idea of such a game; since the plaintiff could not claim copyright protection for these features, a finding of in-

---

screen, file, cursor location, and window information, and a "menu bar" listing available pull down menus;

(4) Provide similar procedures for the user to begin a new outline, or to select an existing outline, from the opening menu;

(5) Use main editing screens for the entry and editing of data;

(6) Select new outline elements and pull-down menus in the same manner;

(7) Permit the user to select directly from the main editing screen certain of the commands displayed on the pull-down menus;

(8) Make available nine pull-down menus (four of whose titles corresponded exactly, and the other five of which have the same or similar functions); and

(9) Generally perform the same outlining functions.

12 U.S.P.Q.2d at 1994-95.

Despite these comparisons, the district court found that there was sufficient difference between the programs to warrant the submission of the infringement question to the jury. First, through analytic dissection it concluded that many of the correspondences resulted from unavoidable, and thus unprotectable, expressions of "concepts . . . fundamental to a host of computer programs." *Id.* at 1995. Thus, the menu screens of both programs could, without infringement, contain options to access files, enter data, edit, and print, which are "essential to the very idea of a computer outline program." *Id.* Similarly, the blue background of both programs, chosen from a limited number of colors, could be seen as a similar response to common considerations of user comfort, monitor design, and the cultural associations of the color. *Id.* at 1995-96. Nor could the use of pull-down menus be protected by copyright, since this idea and expression was, "if not standard, then commonplace in the computer software industry." *Id.* Like the ideas behind them, features so basic to the programs at issue did not qualify for copyright protection.

Second, the district court found that many of the copyrightable aspects of the programs were not sufficiently similar to justify a finding of infringement. For example, although the pull-down windows of both programs used letter abbreviations to select the option, they looked different and described the functions differently. *Id.* at 1996. The opening menus of the programs offered different options, overlapping only in those functions (opening or starting a new outline, accessing an existing outline, accessing the directory, and quitting the program altogether) that did not qualify for copyright protection. *Id.* In addition, the options were described differently, had lettering of different sizes, and were accessible to the user through different menus. *Id.* The court concluded that although the opening screens shared similar ideas, the screens were, as a matter of law, not substantially similar. *Id.*

57. *Brown Bag*, 960 F.2d at 1475-76.

fringement would not stand.[58]

In its own recent inquiry into software copyright infringement, the district court for the District of Massachusetts distinguished *Arnstein*'s two prongs by the relative weights that they gave to legal and factual considerations.[59] First, in establishing that the substantial similarities are the result of actual copying of the plaintiff's work, rather than of the defendant's independent development, " 'substantial similarity' simply means sufficient similarity of a given element of a work to an element in the allegedly infringing work to support a reasoned inference that more probably than not the element was copied from the copyrighted work. . . . This is not similarity in a mixed law-fact sense but . . . in a purely factual [and] evidentiary sense[, as] one of circumstantial evidence of copying."[60] On this issue, expert evidence and a "dissection" of the works in question is relevant.[61]

Second, once the copyrightable elements of the plaintiff's work have been isolated, for the court to find that these elements were copied by the defendant, they must bear such a "substantial similarity" to their counterparts in the defendant's program that "an 'ordinary observer' [would find] that there was 'unlawful appropriation.' "[62] By definition, then, this prong of the test is clearly a mixed question of fact (in determining the degree to which the copyrightable elements are similar) and law (in determining whether their appropriation was unlawful).[63]

Reflecting the increasingly blurred boundary between questions of high technology fact and questions of high technology

58. Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204, 209 (9th Cir. 1988). The similarities involved: the number of moves and opponents allowed per game; the various karate punches and kicks available to the player; the presence of changing background scenes and one referee; the statements of the referee and their depiction; the length of count-down rounds; and the provision of bonus points. *Id.*

59. Lotus Dev. Corp. v. Borland Int'l Inc., 788 F. Supp. 78 (D. Mass. 1992) (plaintiff did not properly frame its allegations that defendant had infringed copyright on elements of user interface).

60. *Id.* at 84.

61. Arnstein v. Porter, 154 F.2d 464, 468 (2d Cir. 1946).

62. *Lotus Dev. Corp.*, 788 F. Supp. at 84 (quoting Concrete Mach. Co. v. Classic Lawn Ornaments, 843 F.2d 600, 608 (1st Cir. 1988)).

63. *Id.*

law, *Computer Associtates* continued the trend towards reliance on expert opinion in evaluating allegations of software copyright infringement. Each of the three steps in the Second Circuit's "abstraction/filtration/comparison" analysis would appear to benefit, if not actually to require, the testimony of an industry insider.

3. *Early Precedent on Copyrightability of Non-Literal Elements*

Between 1978 and 1985, four district courts examined the degree to which competitors could duplicate features of computer programs and/or related documentation. Generally, their decisions characterized elements integral to a program's structure as uncopyrightable "ideas" and permitted their reproduction, even when the plaintiff had arbitrarily chosen the specific configuration of these features (*e.g.*, the sequence of data input) from a wide range of options. By contrast, the defendant's slavish copying or re-creation of non-essential features, or "expressions," was found to constitute copyright infringement.

In *Synercom Technology, Inc. v. University Computing Co.*,[64] the district court confronted the question of whether the plaintiff's copyrighted input formats for its computer program, as evidenced by the design of its manuals and input punch cards, had been infringed by the defendant's nearly identical input manual. Unlike the blank accounting forms of *Baker v. Selden*, which were "[c]ertainly . . . not the subject of copyright,"[65] these input formats expressed "the sequencing of data for simplified access to the computer programs. The formats by their placement of lines, shaded art, and words tell the user what data to place where and how to do it. It communicates the selection arrangements and the sequence. . . . It follows that the formats are copyrightable if the ideas they express are separable from their expression."[66]

The court questioned, though, whether a "separate idea" would remain to be expressed if this sequencing and ordering

64. 462 F. Supp. 1003 (N.D. Tex. 1978).
65. *Id.* at 1011 (citing Baker v. Selden, 101 U.S. 99 (1879)).
66. *Id.* at 1012.

constituted a protectable expression.[67] In holding that the arrangement of input was an unprotectable idea, the court analogized the input format to "[t]he familiar 'figure-H' pattern of an automobile stick [shift, which] is chosen arbitrarily [as the sole configuration, among all possible configurations, to be implemented] by an auto manufacturer."[68] Other auto manufacturers could legally reproduce the same stick-shift pattern, as an idea, on their own production lines, and could provide their customers with their own literature, photographs, or films concerning the shift's operation, "so long as these materials take the form of original expressions of the copied idea (however similar they may be to the first manufacturer's materials) rather than copies of the first manufacturer's expressions themselves."[69]

The necessary/unnecessary distinction has been employed particularly where the defendants have removed any literal similarities in their code through the independent re-creation of the plaintiff's code. In *SAS Institute, Inc. v. S & H Computer Systems, Inc.*,[70] S & H originally provided its programmers with the entire source code of SAS's program.[71] However, the programmers were subsequently isolated from this material and given the tasks of re-creating different portions of the SAS program from specifications that S & H had extracted from the SAS source code.[72] The court found copyright infringement of the plaintiff's

67. *Id.* at 1013.

68. *Id.*

69. *Id.* Since "[s]ubstantial portions of [the defendant's] manuals are verbatim from the [plaintiff's] manual," it was clear to the court that infringement had occurred in the case before it. *Id.* at 1014.

70. 605 F. Supp. 816 (M.D. Tenn. 1983).

71. *Id.* at 825.

72. *Id.* In this second phase, the "individual programmers would be assigned a particular, narrow task, and provided only that information relevant to the task." One programmer, for instance, "was provided a specific statement of the task to be performed by a program module, and was given information [only] as to the ways in which that module of code would interface with the remainder of the [defendants'] product." *Id.* The defendants wrote the source code linking the modules produced by the individual programmers to the remainder of the defendants' product. *Id.*

If the S & H programmers had not previously been exposed to the SAS source code, this process of duplicating a program would have qualified as a "clean room" technique:

statistical analysis software after determining that the similarities between the programs were not dictated by necessity but instead "represent[ed] unnecessary, intentional duplication of expression."[73] In terms of substantial similarity, it appeared that evidence remained of S & H's original approach to duplication: "[i]t certainly cannot be said that 44 specific examples of [direct] copying [of source code] as a matter of law were insubstantial."[74] The court noted that many more examples of copying would undoubtedly have been brought to light had not the defendant "embarked upon its program of destroying and masking evidence and disguising its conduct."[75] Significantly, the copying at issue extended beyond specific lines of code to the structure of the plaintiff's program itself.

However, on very similar facts to those of *SAS*, *Q-Co Industries, Inc. v. Hoffman*[76] declined to find copyright infringement where the defendant had not re-created the plaintiff's source code but instead had duplicated the structure and concept of

---

> Software created by a second party through a "clean room" process is created as follows: The second party forms two teams of programmers. The first team has access to the first party's source code and writes a functional description of the program (*i.e.*, describes input, output and constraints such as size of available memory) without disclosing in any way the method for accomplishing the results. The written functional description (and nothing else) is transmitted to the second team, which then creates software from that functional description. There is no other communication between the two teams and no member of the second team has been exposed to the first party's software. [Since] [m]any contend that developing software from no more than a functional description does not infringe the software's copyright, a properly implemented clean room technique is regarded by most observers in the software industry as a lawful means of creating competing software.

DAVID BENDER, COMPUTER LAW, SOFTWARE PROTECTION § 4.04[1], at 4-22.2(23) n.12.10 (1992) [hereinafter COMPUTER LAW]. *See generally* NEC Corp. v. Intel Corp., No. C-84-20799-WPG, 1989 WL 67434 (N.D. Cal. Feb. 6, 1989) (code developed by clean room process indicates that similarities between parties' programs resulted from constraints of hardware, architecture and specifications).

73. *SAS*, 605 F. Supp. at 825. The court found "as a matter of fact that the expression, and not merely the ideas" of the plaintiff's program were duplicated. *Id.* at 829.

74. *Id.* at 830.

75. *Id.*

76. 625 F. Supp. 608 (S.D.N.Y. 1985).

plaintiff's arrangement of four program modules: "the same modules would be an inherent part of any [tele]prompting program."[77]

Finally, in *E.F. Johnson Co. v. Uniden Corp. of America*,[78] the court found that the computer program integral to the defendant's logic trunked radio system[79] infringed not upon the idea, but upon the expression, of the plaintiff's copyrighted software.[80] The court rejected the defendant's argument that the similarities had been dictated by the goal of compatibility with the plaintiff's system, since this goal could have been achieved without verbatim duplication of the plaintiff's program.[81] The plaintiff, unlike that in *Q-Co*, had introduced an element of originality "to the compilation of standard programming techniques" sufficient to defend its own copyright.[82] The court characterized the extrinsic/intrinsic test of copyright infringement in the computer software context as "problematical," since the intrinsic test

77. *Id.* at 616. Although the defendant had provided his assistant with only that limited amount of information necessary to duplicate the four modules at issue, and had himself written the source code to link those elements of the completed teleprompter program, the court, expanding on *Synercom*'s automotive analogy, found that the "order and organization [of these modules] can be more closely analogized to the concept of wheels for the car rather than the intricacies of a particular suspension system." *Id.* In addition, by contrast to the *SAS* defendant's wholesale copying of a program, the *Q-Co* defendant had been forced to introduce significant modifications into the original program to enable it to run on a different hardware package and in a different computer language. *Id.*

The distinction between the holdings of *SAS* and *Q-Co* may also be attributable to equitable considerations. *See infra* note 132.

78. 623 F. Supp. 1485 (D. Minn. 1985).

79. "A logic trunked radio system is one consisting of mobile radio units, typically installed in motor vehicles such as taxis, police cars, delivery trucks, etc., and 'repeaters,' base stations which receive and transmit signals to and from the mobile radio units. The heart of the [plaintiff's] system is computer software contained in the mobile radios and repeaters." *Id.* at 1487.

80. *Id.* at 1503.

81. *Id.* at 1502 (citing Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521 (9th Cir. 1984) and Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983)).

82. *Id.* at 1499-1500 (citing Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003, 1010 (N.D. Tex. 1978) (computer manual copyrightable even though 30 percent of its content composed of excerpts from material in public domain, since the assembly of old and new parts resulted in original expression).

would be difficult to apply: "any attempt to gauge the 'aura' or 'feel' of a computer program imbedded on a silicon chip is doomed ab initio."[83]

4. *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*

The necessary/unnecessary distinction made by the early cases in software copyright was taken to its extreme by the Third Circuit in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*[84] a "case of first impression in the courts of appeals."[85] *Whelan* restricted the noncopyrightable "idea" of a program to that program's "purpose or function," thereby endorsing the protection of "everything that is not necessary to the expression of the idea,"[86] including the program's "structure, sequence and organization."[87] This narrow definition of "idea" has the potential to restrict development of rival programs compatible with those of an industry pioneer, because a program's general architecture, in addition to its actual source code, would be protected as expression.

Whelan addressed the alleged infringement of a computer program designed to streamline the business operations of dental laboratories.[88] The court noted that the facts of the case before it, and various commentaries on software development, indicated that the cost- and labor-intensive aspect of software development was not the coding of the program into computer language but creating the program's "structure and logic."[89]

The plaintiff's ownership of the copyright and the defendant's

---

83. *Id.* at 1501 n.16. The court reduced the "primary inquiry" in infringement cases to "whether the stated objective [of the programs at issue] can be accomplished in only one or a few ways." *Id.*

84. 797 F.2d 1222 (3d Cir. 1986), *cert. denied,* 479 U.S. 1031 (1987).

85. *Id.* at 1224.

86. *Id.* at 1236.

87. *Id.*

88. The "significant [and, apparently, specialized] bookkeeping and administrative tasks" to be completed by such laboratories include: the registration and processing of orders for equipment; maintenance of inventory; continual updating of customer lists; and administration of invoicing, billing, and accounts receivable. *Id.* at 1225.

89. *Id.* at 1231.

access to this program were undisputed.[90] The plaintiff did not allege that the source or object codes of its software had been copied;[91] indeed, the two versions of the program at issue had been developed in different computer languages.[92] Thus, the only issue before the Third Circuit was whether[93] there was substantial similarity between the non-literal structure of the two works in question sufficient to indicate copyright infringement.[94]

The court affirmed the district court's collapsing of the "extrinsic/intrinsic" test advanced by *Arnstein v. Porter*.[95] "The ordinary observer test, which was developed in cases involving novels, plays, and paintings, and which does not permit expert testimony, is of doubtful value in cases involving computer programs on account of the programs' complexity and unfamiliarity to most members of the public."[96] In addition, it made little sense to require the finder of fact applying the intrinsic test to disregard the same expert testimony received while determining the extrinsic test.[97] Instead, the court adopted "a single substan-

90. *Id.* at 1232.

91. *Id.* at 1233.

92. Elaine Whelan, the programmer retained by Jaslow Laboratories (JL) to develop management software for dental laboratories, wrote the program Dentalab in EDL (Event Driven Language) for compatibility with IBM Series One computers. *Id.* at 1226. After Rand Jaslow of JL had produced a similar program, Dentcom, in BASIC, he gave notice of termination of the joint marketing agreement then in force between JL and Whelan Associates (Whelan), warning that JL considered itself the exclusive marketer of Dentalab. Jaslow and other individuals subsequently formed a company to market the Dentcom program in addition to (and as a new version of) Dentalab. However, Whelan continued to market Dentalab. *Id.* at 1226-27.

After JL sued Whelan in the Court of Common Pleas of Montgomery County, Pennsylvania, for misappropriation of trade secrets, Whelan brought a copyright infringement suit against JL in the United States District Court for the Eastern District of Pennsylvania. The District Court held that the structure, if not the literal code, of JL's Dentcom had infringed Whelan's copyright in Dentalab. *Id.* at 1228-29 (citing Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 609 F. Supp. 1307, 1321-22 (E.D. Pa. 1985)).

93. A finding of infringement was subject to the defendant's ability to demonstrate that it had developed its program independently and/or through material in the public domain.

94. 797 F.2d at 1232-33.

95. *Id.* at 1232 (citing Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 609 F. Supp. 1307, 1321-22 (E.D. Pa. 1985)).

96. *Id.* at 1232.

97. *Id.* at 1232-33.

tial similarity inquiry according to which both lay and expert testimony would be admissible."[98]

Noting that *Baker v. Selden* denied copyright protection to blank forms precisely because these aids were so closely linked to "the end sought to be achieved by Selden's book," (*i.e.*, the implementation of an uncopyrightable accounting system), *Whelan* enunciated a new version of the idea/expression distinction for the software context:[99] "*the purpose or function of a utilitarian work would be the work's [non-copyrightable] idea, and everything that is not necessary to that purpose or function would be part of the [copyrightable] expression of the idea* . . . . Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea."[100] Under this analysis, the first developers of a program for a specialized market could obtain a copyright for all non-literal elements that could have been otherwise structured, even if all other arrangements would be comparatively inefficient. Moreover, this approach would effectively abolish the usefulness of "clean room" procedures, by which competitors replace the literal code, but retain the architecture, of original programs.[101]

Whelan rejected *Synercom's* implication that sequence and

---

98. *Id.* at 1223. The court noted that in both E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485, 1493 (D. Minn. 1985) and Hubco Data Prods. Corp. v. Management Assistance Inc., 2 Copyright L. Rep. (CCH) ¶ 25,229 (D. Idaho Feb. 3, 1983) infringement determinations had been made entirely on the basis of expert testimony, rather than that of a reasonable, average, or lay observer. *Id.*

99. *Id.* at 1238. "Since it may be impossible to discuss the purpose or function of a novel, poem, sculpture, or painting, the rule may have little or no application to cases involving such works." *Id.*

100. *Id.* at 1236. The Third Circuit upheld the district court's determination that "[t]he 'expression of the idea' in a software computer program is the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating, and producing useful information either on a screen, print-out or by audio communication." *Id.* at 1239.

The court cited as support *SAS Institute*'s reference to the "copying of the organization and structural details" into the offending program. SAS Inst., Inc. v. S & H Computer Sys., 605 F. Supp. 816, 830 (M.D. Tenn. 1985), *quoted in Whelan*, 797 F.2d at 1239.

101. J. Diane Brinson, *Copyrighted Software: Separating the Protected Ex-*

form were less copyrightable in the computer context than in others.[102] To counter Judge Higginbotham's suspicion that a program's sequence and organization could not be seen as expressions of any separable idea, *Whelan* reiterated that "the idea [being expressed] is the efficient organization of a dental laboratory."[103] A program designed for this purpose was distinguishable from software for general office efficiency or other ends, and itself implemented only one of a number of possible approaches towards this goal. Thus, the Third Circuit extended copyright protection to the structure, sequence and organization, or "non-literal" elements, of the program at issue, including "the manner in which the program operates, controls, and regulates the computer in receiving, assembling, calculating, retaining, correlating and producing useful information either on a screen, print-out or by audio communication."[104]

Recognizing the controversial nature of its approach to copyrightability, the court attempted to pre-empt various lines of anticipated criticism. First, it indicated that no sympathy should be spared for a software developer who after investing significant

---

pression from Unprotected Ideas, A Starting Point, 29 B.C. L. REV. 803, 852-53 (1988). *See supra* note 72.

102. 797 F.2d at 1240. *See also* COMPUTER LAW, *supra* note 72, § 4.01, at 4-4.4. (*Synercom*, which involved the copyright on the structure, sequence and organization of a format form rather than of a program itself, does not contradict *Whelan* and its progeny). Bender has criticized *Synercom* itself for "its overly narrow choice of the fully detailed format as the idea, [which] departs from logic and is contrary to well-established copyright precedent." *Id.* § 4.04[1], at 4-22.1.

103. 797 F.2d at 1240. The court observed that structure and organization might rise to the level of uncopyrightable "idea" if the program's very idea or purpose was to employ that structure or organization. *Id.* at 1238.

104. *Id.* at 1239 (citing *Whelan Assocs.*, 609 F. Supp. at 1320). The specific similarities leading to a finding of copyright infringement in *Whelan* were found in: (1) the file structures in the two programs (which required, and ordered in a particular fashion, specific information); (2) the screen displays of the programs; and (3) the five subroutines of both programs. *Id.* at 1243-46. Significantly, the substantial similarity analysis was not affected by the fact that screen displays are covered by a different copyright category than is software, or by the fact that the plaintiff had not argued that the displays infringed its work. *Id.* at 1244. Nor did the court accept the argument that substantial similarity could be determined only after a comparison of a majority of the structure of the two works: comparison of software, as of literary works, would proceed on a qualitative, not quantitative, basis. *Id.* at 1245.

cost and effort managed to re-create a program's protectable non-literal elements (through "clean room" techniques or otherwise) without copying the source code or object code of the original: not only would the cost to the infringer be irrelevant,[105] but even the "approximation" of the structure of a popular program would afford her "a significant advantage over competitors."[106]

Second, the court evinced a willingness to sacrifice predictability and ease of determination in order to expand copyright protection beyond software's "literal codes" to its non-literal elements.[107] Nor would the new test retard the evolution of the field. *Whelan* specifically rejected the argument that software development is a uniquely self-cannibalizing enterprise, in which each creation necessarily incorporates elements of its predecessors: "We are not convinced that progress in computer technology or technique is qualitatively different from progress in other areas of science or the arts."[108]

Finally, the court dismissed as irrelevant the CONTU Report's recommendation that copyright protection be extended to software[109] as the Report did not propose limiting such protection to literal elements of software,[110] and since in any event the Report "cannot be a substitute for legislative history in this case."[111]

---

105. To this extent, the Third Circuit anticipated Feist Publications, Inc. v. Rural Tel. Serv. Co., 111 S. Ct. 1282, 1290 (1991), which rejected the "sweat of the brow" doctrine that would have afforded protection to factual compilations whose "author" had invested substantial time in their assembly.

106. 797 F.2d at 1237.

107. *Id.* at 1238.

108. *Id.*

109. *Id.* at 1241 (citing CONTU REPORT, *supra* note 12, at 1).

110. The court noted that the CONTU Report extended copyright protection beyond source code and object code to flow charts. *Id.* (citing CONTU REPORT, *supra* note 12, at 21).

111. *Id.* at 1241-42. Observing that the binding effect of the CONTU Report as a "surrogate legislative history" has been predicated on the absence of Congressional alterations to or committee reports on its recommendations, the Third Circuit noted that "the only statutory provision relevant to this case is § 102(b) [precluding copyrightability of ideas] . . . to which no changes were made as a result of the CONTU Report." *Id.*

5. *The Initial Response to* Whelan

a. *Criticisms*

Although some courts have followed *Whelan*,[112] its "purpose"-based analysis has by no means received unanimous support. For example, the Fifth Circuit, citing *Synercom*'s gearstick analogy, refused to apply the *Whelan* test to protect the sequence and organization of cotton-marketing software.[113] The record reflected that the sequence and organization were largely dictated by the structure of the cotton market.[114]

In *Autoskill, Inc. v. National Educational Support Systems Inc.*,[115] the alleged copyright infringement concerned the development of a program for the testing, diagnosis and treatment of reading skills. The defendant's access to plaintiff's version was undisputed. Comparing the programs' equivalent division of functions (into three main sections), progression of lessons (from simple to complex), training approaches, and feedback methods, the district court for the District of New Mexico found the works "substantially similar [in their] important aspects."[116]

To determine whether plaintiff's program for testing, diagnosis and training of reading skills was itself copyrightable, the court looked to Learned Hand's "increasing abstractions" test.[117] It determined that the program's general purpose and orientation towards three distinct reading-level subtypes identified by re-

112. *See, e.g.*, Bull HN Info. Sys., Inc. v. American Express Bank Ltd., No. 88 Civ. 2103, 1990 WL 48098 (S.D.N.Y. Apr. 6, 1990) (deferring to trial the question of whether functions of plaintiff's compiler program were duplicated by differently-structured rival programs, *i.e.*, whether plaintiff's program was copyrightable under *Whelan*); Dynamic Solutions, Inc. v. Planning & Control, Inc., 646 F. Supp. 1329 (S.D.N.Y. 1986) (agreeing with *Whelan's* categorization of inessential structural similarities as expressions); Healthcare Affiliated Servs., Inc. v. Lippany, 701 F. Supp. 1142 (W.D. Pa. 1988) (infringement claim rejected where plaintiff did not prove that choices made in development of program at issue did not rise to level of structure, sequence or organization).

113. Plains Cotton Co-Op. v. Goodpasture Computer Serv., Inc., 807 F.2d 1256, 1262 (5th Cir.), *reh'g denied*, 813 F.2d 407 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987).

114. *Id.*

115. 793 F. Supp. 1557 (D.N.M. 1992).

116. *Id.* at 1560.

117. *Id.* at 1566 (citing Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931)).

searchers in material available to the general public were uncopyrightable ideas. Autoskill could copyright, as expressions, its original training methods, but could not protect features that were "dictated by the use of the English language"[118] or "such standard devices in reading programs that they cannot be considered in analyzing substantial similarity."[119] After examining both expert and lay evidence, as well as exhibits presented to the court, the court found a "substantial similarity between the [defendant's program] and the most important protectable aspects of Autoskill's [p]rogram."[120]

The court rejected "[t]he *Whelan* court's approach, [which,] although a temptingly simplistic and bright line approach, cannot account for the reality that many ideas may exist in a given work. . . . Adopting the *Whelan* rule would also put a damper upon the important goal of encouraging others to build upon the ideas conveyed in a work."[121] Significantly, the court cited the complexity of the programs as one factor in its decision not to apply the "total concept and feel" test for copyright infringement, which "is more appropriate when evaluating simplistic works where unanalytic evaluation is appropriate."[122]

Whelan has been hailed by one commentator for bringing to the idea/expression controversy an emphasis on the actual process of software development, from the idea of a program's purpose to the expression of the source code that implements that purpose.[123] Yet even this evaluator has attacked the Third Circuit's analysis as technically vague and imprecise: it "adopt[s] a

118. Such features included the thirteen categories of vowel and consonant combinations used in the program. *Id.* at 1560.

119. For example, silent sentences and silent paragraphs. *Id.*

120. *Id.* at 1569.

121. *Id.* at 1566.

122. *Id.* at 1570 (citing 3 NIMMER & NIMMER, *supra* note 30, § 13.03[A], at 13-37 to 13-41). This test, advanced in Roth Greeting Cards v. United Card Co., 429 F.2d 1106 (9th Cir. 1970), relies only on the impressions of the fact finder, and not on expert testimony. "Because the court did not operate the programs, in order to utilize this test I could only rely upon a few photos of selected screen displays and a logic flow chart. Such a determination would not be meaningful. I would need to rely upon the explanations and impressions of the expert witnesses which [would be] inappropriate" for the application of the "total concept and feel" test. *Autoskill*, 793 F. Supp. at 1570.

123. Bender, *supra* note 18, at 3-4.

stilted view of what constitutes the 'idea' to be distinguished from expression under copyright law."[124]

Generally, legal scholars have elaborated on *Autoskill*'s criticisms of *Whelan*. First, the Third Circuit's restriction of "idea" to the underlying purpose of a computer program has been seen as "so abstract that it renders the idea/expression distinction fundamentally meaningless."[125] One program can have many different levels of abstraction, each with its own idea: to protect as expression everything but the most general of these ideas, that of the program's purpose, effectively relegates all subsidiary ideas, including those regarding the organization of each of the program's sublevels, to the domain of copyrightable expression.[126]

Second, *Whelan* had inferred from the Copyright Act's protection of compilations that the structure, sequence and organization of software itself were copyrightable:[127] such protection "would provide the proper incentive for programmers by protecting their most valuable efforts, while not giving them a stranglehold over the development of new computer devices that accomplish the same end."[128] However, the Supreme Court's later decision in *Feist Publications, Inc. v. Rural Telephone Service Co.*,[129] which declined to extend copyright protection to the "white pages" of a telephone directory, resoundingly rejected the "sweat of the brow" doctrine, under which "copyright has been considered a reward for the hard work that went into compiling facts."[130] Thus, to the extent that *Whelan* can be said to have

---

124. *Id.*

125. Marc T. Kretschmer, *Copyright Protection for Software Architecture: Just Say No!*, 1988 COLUM. BUS. L. REV. 823, 837 (1988).

126. *Id.* at 839.

127. "Title 17 U.S.C. § 101, defines 'compilation' as 'a work formed by the collection and *assembling* of preexisting materials or of data that are selected, *coordinated*, or *arranged* in such a way that the resulting work as a whole constitutes an original work of authorship,' and it defines 'derivative work,' as one 'based upon one or more preexisting works, such as . . . *abridgement*, *condensation*, or any other form in which a work may be *recast*, transformed or adapted.' " Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1239 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

128. *Id.* at 1237.

129. 111 S. Ct. 1282 (1991).

130. *Id.* at 1291 (citing Jeweler's Circular Publishing Co. v. Keystone Publishing Co., 281 F. 83 (2d Cir.), *cert. denied*, 259 U.S. 581 (1922) (copyright-

relied on or endorsed that doctrine, it has been overruled.[131]

Third, it has been suggested that the Third Circuit, incensed by the defendant's blatant duplication of program elements, manipulated the idea/expression test in order to find copyright infringement.[132]

Fourth, the broad protection that *Whelan* offers to the structure, sequence and organization of programs "may force later competitors to produce products that are markedly different from the copyrighted product to avoid possible threats of copyright lawsuits. . . . This is a substantial barrier to a new competitor entering a marketplace."[133] Moreover, requiring programmers to develop new techniques to replace those protected by *Whelan* "duplicates effort and wastes resources."[134]

---

ability of compilation dependent on labor expended by author of work, and not on public or non-public nature of collected material itself)).

131. *Id.* at 1295 ("[T]he 1976 revisions to the Copyright Act leave no doubt that originality, not 'sweat of the brow,' is the touchstone of copyright protection in directories and other fact-based works.").

See, e.g., CMAX/Cleveland, Inc. v. UCR, Inc., 804 F. Supp. 337, 352 (M.D. Ga. 1992) (declining to adopt *Whelan* analysis since the "sweat of the brow" rationale underlying *Whelan* test has been rejected by the Supreme Court in *Feist* and since *Whelan* test is inadequate to address complexity of computer programs).

Cf. supra note 105 (*Whelan* rejected "sweat of the brow" doctrine as applied to infringers).

132. Kretschmer, *supra* note 125, at 838. *See also* John M. Conley & David W. Peterson, *The Role of Experts in Software Infringement Cases*, 22 GA. L. REV. 425, 434-35 (1988) (proposing that the *SAS* and *Whelan* courts found plagiarism because their "evaluation of the defendants' conduct as fundamentally inequitable influenced their views of the extent of copyright protection available to software authors," while attributing difference in the *Q-Co* result to the court's focus on absence of evidence of defendant's pervasive use of plaintiff's program).

133. William Wright, *Litigation as a Mechanism for Inefficiency in Software Copyright Law*, 39 UCLA L. REV. 397, 418 (1991).

134. Peter Spivack, *Does Form Follow Function? The Idea/Expression Dichotomy in Copyright Protection of Computer Software*, 35 UCLA L. REV. 723, 752 (1988).

"[T]he *Whelan* rule distends copyright protection, placing off-limits alternative and improved means of expression and thereby upsetting the uneasy balance which copyright attempts to maintain by preventing free riders from ripping off creative expression while not stifling others from improving or extending that expression." Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1025 (N.D. Cal. 1992) (granting Microsoft's motion for partial summary judgment against infringement claim, since none of alleged similari-

This restriction stifles the developers' culture of building on each others' works and restricts the dissemination of efficient subprograms.[135]

Whelan's insistence on divergence among competitive products burdens users unwilling to learn, or to be retrained for, significantly different software to accomplish a given task. By reducing incentives to improve the product and/or to lower its

---

ties in Microsoft Windows came close to being identical copy of corresponding features in Apple products, and since each of the elements is subject to at least one limiting doctrine, such as *scenes a faire* or merger) (citing Computer Assocs. Int'l v. Altai, Inc., Nos. 91-7893, 91-7935, 1992 U.S. App. LEXIS 14305 (2d Cir. June 22, 1992)).

135. Spivack, *supra* note 134, at 752-55; Kretschmer, *supra* note 125, at 841-42. Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1082-83 (1989); Vance F. Brown, *The Incompatibility of Copyright and Computer Software: An Economic Evaluation and a Proposal for a Marketplace Solution*, 66 N.C. L. REV. 977, 997-98 (1988).

The same criticisms have been leveled against Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990); *see* discussion *infra* parts V.B.1-B.3. *See also* Karen S. Kovach, *Computer Software Design: User Interface — Idea or Expression?*, 60 U. CIN. L. REV. 161, 185-86 (1991); Gregory J. Ramos, Note, *Lotus v. Paperback: Confusing the Idea-Expression Distinction and Its Application to Computer Software*, 63 U. COLO. L. REV. 267, 290 (1992) (predicting that clone computers would no longer be manufactured, and that "either IBM would be awarded a complete monopoly on microcomputers [using its proprietary Read Only Memory-Basic Input/Output System (ROM-BIOS) component], or manufacturers would inundate the market with incompatible units."). *But see* N.Y. TIMES, Aug. 20, 1992, at D3 (Lotus's jubilant full-page newspaper advertisement in the wake of Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (D. Mass. 1992), stating that "[w]e sued to protect our intellectual property rights. And in winning we've helped preserve an environment in which independent software developers can freely develop innovative new products without fear that their creative work will be stolen. We sued to protect innovation, not to stifle it.").

Two scholars of the Law and Economics movement have suggested that the debate over the copyright protection of software's "look and feel"

> will be resolved not by the semantics of the words 'idea' and 'expression' but by the economics of the problem and, specifically, by comparing the deadweight costs of allowing a firm to appropriate what has become an industry standard with the disincentive effects on originators if such appropriation is forbidden. The probability that a particular display format will become the industry standard is small; presumably there are significant returns over and above copyright to a firm that achieves such a position; and the narrowly expressive aspects of the display are protected, thus limiting the effect of free riding.

William Landes & Richard Posner, *An Economic Analysis of Copyright Law*, 28 J. LEGAL STUD. 345, 352 (1989).

price, *Whelan* decreases competition, thereby entrenching the established market leaders and further encouraging market (and software) inefficiency.[136]

Finally, the *Whelan* test may be ignored by jurors who, instead of applying its "purpose" criterion, may be "far more likely to be swayed by the appearance that a defendant has copied from a copyrighted piece of software."[137]

b. *Extensions of* Whelan

Although controversial, *Whelan* has become a factor in the "third wave" of software copyrightability cases, which involve protection of user interfaces.[138]

Broderbund Software, Inc. v. Unison World[139] broadened *Whelan*'s protection of a program's structure, sequence and organization to include screen displays. *Broderbund* found the menu screens, input formats, and sequencing of screens in plaintiff's "Print Shop" program copyrightable because a nonparty marketed a comparable product offering substantially the same functions but through a user interface with a significant structure and organization.[140] "[T]he separable idea of 'Print Shop' is the creation of greeting cards, banners, posters and signs that contain infinitely variable combinations of text, graphics, and borders. A rival software publisher is completely free to market a program with the same underlying idea, but it must express the idea through a substantially different structure."[141]

---

136. Wright, *supra* note 133, at 419; Spivack, *supra* note 134 at 753-54.

137. Wright, *supra* note 133, at 418-19.

138. Bender, *supra* note 18, at 2-3. "The third wave of cases represent a bold attempt by software developers to focus on those aspects of the use of their software that they view as commercially important. Having identified those aspects, they have sought, by use of copyright, to exclude others from using those aspects." *Id.*

139. 648 F. Supp. 1127 (N.D. Cal. 1986).

140. *Id.* at 1132-33.

141. *Id.* at 1133. *See also* Healthcare Affiliated Servs., Inc. v. Lippany, 701 F. Supp. 1142, 1152 (W.D. Pa. 1988) (entering summary judgment against plaintiffs' copyright infringement claim, where plaintiffs had not demonstrated that methodologies of their hospital management software comprised the "structure, sequence and organization" of a computer program within the meaning of copyright law and of *Whelan*).

Broderbund found that *Krofft*'s analytic dissection, or side-by-side comparison of similarities, controlled the substantial similarity inquiry. Noting that *Whelan* did not specify the role of analytic dissection in determining substantial similarity, the district court suggested that "an integrated test involving expert testimony and analytic dissection may well be the wave of the future in this area."[142]

Yet *Digital Communications v. Softklone Distributing*[143] noted that *Whelan* had held only that the duplication of a program's screen displays might be evidence of copying, and that the Third Circuit had in fact supported the separate copyrighting of screen displays as "audiovisual works."[144] *Softklone* thus denounced *Broderbund*'s protection of screen displays as based on "an overexpansive and erroneous reading of *Whelan*."[145] In concluding that the plaintiff's separate copyright on a "status screen"[146] had

142. 648 F. Supp. at 1136.

143. 659 F. Supp. 449 (N.D. Ga. 1987).

144. *Id.* at 455 (citing Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986)).

145. 659 F. Supp. at 455. *See also* Manufacturers Technologies, Inc. v. CAMS, Inc., 706 F. Supp. 984, 993 (D. Conn. 1989) (*Broderbund* overextended scope of copyright protection of screen displays by misinterpreting *Whelan*'s statement that inferences of copying could be drawn from similarity of screen displays).

146. The "status screen" screen display [of plaintiff's communication program], which appears immediately following the "boot-up" or sign-on screen display, contains in its upper portion an arrangement and grouping of parameter/command terms under various descriptive headings. Next to each of the parameter/command terms are values, either numerical or verbal. The value of each parameter/command reflects the value at which the program is operating and is either selected by the user or by the computer program ("default settings"), *e.g.*, the number 300 next to the "SPeed" parameter/command indicates the byte or baud rate at which the computer program is communicating with other computers. Two letters of each parameter/command term are capitalized and highlighted. By typing those two letters, the user can effectuate that specific command.

The lower portion of the status screen display, excluding the bottom line, called the "window," can display a wide variety of text, including anything the user might wish to cause to appear there. Upon typing in a "HElp" command, the user can call up into the "window" a list of all the [relevant] parameter/command terms. The list of terms is arranged in four alphabetical groupings. Because of the size of the list, all the terms cannot be displayed in the window at one time. By

been infringed, *Softklone* itself specifically upheld *Whelan*'s conclusion that blank forms could be copyrightable if sufficiently innovative.[147] Unlike *Synercom*'s non-infringing duplication of format cards, *Softklone* involved the copying of the "arrangement, headings, capitalization and highlighting" of the elements on such a screen display.[148] The court identified as uncopyrightable "ideas" the use of a screen to display the program's status, the activation of commands by the typing of two symbols,[149] and the use of commands to drive the program: "All of these elements relate to how the computer program receives commands or instructions from the user and how operationally the computer program reflects the results of those commands."[150] By contrast, other elements, such as the sequence of entry of the parameter/command terms and the highlighting and capitalization

---

pressing the "enter" key on the keyboard when the first portion of the term is shown in the window, the remainder of the terms will appear.

The bottom line of the status screen is the "command" line. On this line, the user can enter "commands" or instructions to the computer to change the values at which it operates. After entering a command changing the operation of the program, the change is then reflected by a change in the value next to the corresponding parameter/command term in the upper portion of the screen. For example, if the user wishes to change the byte or baud rate (speed) of the program to 1200, he can type and enter the two letter symbol for the byte or baud rate command along with the rate he desires, *e.g.*, "SP 1200," the computer will then operate at a 1200 byte or baud rate, and the number "1200" will appear in the upper portion of the status screen next to the parameter/command term "SPeed."

659 F. Supp. at 452-53.

147. *Whelan*, 797 F.2d at 1242-43.

148. 659 F. Supp. at 460.

149. Activation of the status screen is accomplished by:

the user typing on the bottom line of the screen two symbols, which correspond to a particular command, usually followed by a value. The computer then affects [sic] a change in the operation of the program based upon the particular command and value involved and reflects the changed status of the program on the upper portion of the status screen. As an example, if the user wishes the program to operate on a certain speed, e.g. 1200, the user can type on the bottom line of the status screen the two symbols "SP" followed by "1200." The computer will then operate at a speed of 1200 and that fact will be reflected by the appearance on the upper portion of the status screen of "1200" next to the term "SPeed."

Id. at 458-59.

150. *Id.*

of two specific letters of such terms, were copyrightable since they were unrelated to the actual operation or function of the status screen.[151] The plaintiff's choices of the arrangements, patterns and groupings of status screen elements were similarly copyrightable as expressions, since they were "clearly not necessary to the idea of a status screen" and, far from being dictated by the nature of the screen itself, had been selected from among "an almost infinite number" of options.[152]

As indicated in *Manufacturers Technologies, Inc. v. CAMS, Inc.*,[153] the Copyright Office has thrown *Softklone*'s validity into doubt by refusing to accept registrations of screen displays separately from registrations of the underlying software.[154] In response, *CAMS* itself created a "legal fiction" that the single registration of a computer program would also register "the screen displays or user interface of that program, to the extent that each contains copyrightable subject matter."[155]

---

151. *Id.* at 460.

152. *Id.*

153. 706 F. Supp. 984 (D. Conn. 1989).

154. *Id.* at 991 (citing 36 Pat. Trademark & Copyright J. (BNA) 155 (1988)). Screen display registrations approved before this decision, however, remain valid. *Id.* at 991 n.13.

155. *Id.* at 993. The court explained that this method would preserve the approach of *Softklone* "by focusing on the copyrightable expression in each type of registration and avoiding the mistake of identifying a program's idea of a particular screen display or some element therein. It recognizes that a computer program and its screen displays are, for copyright purposes, fundamentally distinct [and thus] conforms to the realities of Copyright Office registration procedures." *Id.*

CAMS refused to copyright the plaintiff's "programming conventions" in its program for estimating the costs of machining manufactured parts. The court found that because the locations of various items on the screen had been selected from a very limited range of options, allowing copyright protection for those items would preclude all future use of the substance. *Id.* at 995. Similarly, the plaintiff's "internal method of navigation" to select or change functions or entries was found to be constrained by the hardware on which the program operated. *Id.* However, plaintiff's method of selecting "what should be made part of the status report, arrangement of the terms therein, assignment of numbers to specific operations/departments and tools, and coordination in the manner of building on the status report as the user progresses through to various steps" was copyrightable, as an expression of "the idea of apprising the user of the status of one's efforts in cost-estimating a part." *Id.* at 996.

III. *COMPUTER ASSOCIATES INTERNATIONAL, INC. V. ALTAI, INC.*: TOWARDS A NEW SUBSTANTIAL SIMILARITY ANALYSIS

A. *Background*

1. *Parties and Products*

Computer Associates' (CA's) program CA-SCHEDULER, first available to customers in 1982,[156] created and implemented schedules for jobs to be run on IBM's System 370 family of mainframe computers.[157] Depending on its size, each of the System 370 computers contained one of three operating systems: DOS/VCE, MVS, or CMS.[158] Operating systems, which control "the resources of the computer, allocating those resources to other programs as needed,"[159] are mutually incompatible to the extent that software written for one operating system generally cannot be run on another.[160]

However, a key sub-program, ADAPTER, saved CA-SCHEDULER's users the cost and effort of purchasing and

---

156. Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 554 (E.D.N.Y. 1991), *aff'd in part, vacated in part*, 982 F.2d 693 (2d Cir. 1992).

157. *Id.* at 550-52.

> Programs that run on the same computer and operating system can be divided into three types. The first, resident programs, reside in the computer's memory where they run continuously. CA-SCHEDULER and Altai's ZEKE are resident programs. A second type, batch programs, are scheduled, started, and run to completion of a particular job. The third group, server programs, are resident programs used by other programs to perform a service. ADAPTER and OSCAR are both classified as server programs.

Id. at 549.

158. "DOS/VSE [is used] with the small-to-medium-size [System 370] computers; MVS [is used] for larger computers and CMS for interactive computers." *Id.* at 550.

159. *Id.* "Operating system software interacts with whatever other programs are being used or 'executed' by the computer, providing computer resources such as processors, memory, disk space, printers, tape drives, etc. for the other programs that need them through what are often referred to as 'system calls.' " *Id.*

160. "For [the] interaction [between the operating system software and programs being executed] to occur properly, the other programs must be compatible with the operating system software in use on the computer, *i.e.*, they must be able to exchange information precisely and accurately with the operating system to interact with those computer resources." *Id.*

learning different versions of the same application to run on IBM mainframes employing different operating systems, or of changing the operating systems themselves to avoid buying new programs. As an "interface" or "compatibility component," ADAPTER translated into the appropriate operating system language the "system calls" issued by CA-SCHEDULER's "task-specific" portion, whose own operation did not distinguish among operating systems.[161] ADAPTER, which had also been developed by CA[162] but which could not be used independently of CA-SCHEDULER,[163] also streamlined CA's approach to software development.[164]

Also in 1982, Altai first marketed its own job scheduling program, ZEKE, for use exclusively with the VSE operating sys-

---

161. *Id.* at 551.

162. "ADAPTER was designed and written in 1979 for use with a group of CA's programs called the DYNAM line. CA included a separate copyright notice on ADAPTER with 1979 as the date of first publication. Since 1979, there have been several revisions and changes to [the] ADAPTER code. CA never registered ADAPTER in the copyright office as a separate computer program." *Id.* at 552.

163. *Id.*

164. *Id.* The tripod approach to operating system compatibility also involves drawbacks for the developer, who must

> develop and maintain many different versions of the same program. Changing the program or adding new features requires modifications to all versions. . . . Marketing multiple versions of each product also increases the work required for development and maintenance. . . . This is a cumbersome way to develop and maintain products that run on multiple operating systems.

Id. at 551.

By contrast, the division of programs into task-specific and interface components benefits both the developer and the end-user.

> First, to adapt a program to a new version of an existing operating system or an entirely new operating system, the developer need only modify the interface component. . . . Second, revisions to the task-specific component of the program to correct problems or add features will not affect the program's ability to run under all operating systems, as long as the changes that request operating system services use the interface component. This method is an efficient and effective way to develop and maintain an application or system program that runs in many operating systems. . . . [T]he end user can switch operating systems and still use a given application or system program without apparent differences.

Id.

tem.[165] In late 1983, James P. Williams, himself both a former product manager for CA and Altai's sole programmer (and, at the time of the litigation, Altai's president)[166] sought out CA programmer and former co-worker Claude F. Arney III to help Altai design a MVS-compatible version of ZEKE.[167] Williams had not worked on the development of, or seen the codes of, CA-SCHEDULER or ADAPTER, and did not know that ADAPTER was a component of CA-SCHEDULER.[168] Arney, though, had helped CA develop the VSE version of ADAPTER[169] and had breached his employment agreements by taking to Altai copies of the source code for ADAPTER's VSE and MVS versions.[170]

Williams, who was unaware that Arney possessed the ADAPTER code, proposed creating an MVS-compatible ZEKE by modifying 30 percent of the VSE version of ZEKE.[171] Ultimately, though, he accepted Arney's suggestion that a "common system interface" component be added to ZEKE.[172]

Arney took a month to design a VSE version and another three months to produce a MVS version of this program, OSCAR.[173] Unbeknownst to Williams and the other Altai employees, Arney derived 30 percent of OSCAR codes from his pirated copies of CA's ADAPTER codes.[174] From 1985 to August 1988, Altai incorporated the first generation of OSCAR programs, OSCAR 3.4, into its programs ZEKE, ZACK and ZEBB.[175]

In late July 1988, CA became aware of the apparent appropriation of its ADAPTER code; in response, it copyrighted CA-SCHEDULER and brought a copyright and trade secret misap-

165. *Id.* at 552.
166. *Id.* at 553.
167. *Id.*
168. *Id.*
169. *Id.*
170. *Id.*
171. *Id.*
172. The district and circuit courts' opinions do not indicate the extent to which Arney and Williams discussed the similarity of this approach to that of CA's ADAPTER.
173. *Id.* at 554.
174. *Id.*
175. *Id.*

propriation action against Altai.[176] Only at this point did Altai discover Arney's misappropriation of CA's source code. After Arney admitted to Williams the extent of the copying, Williams, who had not seen the ADAPTER code, had Arney specify the sections of OSCAR 3.4 that had been derived from ADAPTER. Williams then gave Altai's eight other programmers, none of whom had worked on any version of OSCAR, the task of re-creating those sections of OSCAR 3.4 to match ZEKE's requirements.[177] Six months later, in mid-November 1989, Altai released OSCAR 3.5 and provided it to purchasers of OSCAR 3.4 as a "free upgrade."[178]

2. *Procedural History*

CA brought suit in the United States District Court for the District of New Jersey in August 1988.[179] CA alleged that in developing the programs OSCAR 3.4 and OSCAR 3.5, Altai had misappropriated CA's trade secrets and had infringed on CA's copyright in ADAPTER.[180]

In March 1989, the parties stipulated the transfer of the action to the Eastern District of New York.[181]

After a six-day trial before Judge Pratt, who was sitting in the district court by designation,[182] the district court entered judgment on August 12, 1991.[183]

The court held that OSCAR 3.5 did not infringe CA's copy-

176. *Id.*

177. *Id.* "The process of rewriting involved first determining what operating system services were needed by ZEKE; this list was formulated by reference to the version of ZEKE marketed before OSCAR was developed. Williams provided a brief description of each service to a programmer through the parameter lists, and told the programmer to write the appropriate code for obtaining that service through the operating system." *Id.*

178. *Id.*

179. *Id.* at 549.

180. "CA is a Delaware corporation, with its principal place of business in Garden City, New York. Altai is a Texas corporation, doing business primarily in Arlington, Texas." Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 698 (2d Cir. 1992).

181. *Id.* at 700.

182. 775 F. Supp. at 549.

183. *Id.*

right on ADAPTER.[184] Although Altai conceded that it had misappropriated CA's trade secrets,[185] the court held that § 301(a) of the Copyright Act preempted CA's trade secret claim.[186] After an extensive review of the parties' damage analy-

---

184. *Id.* at 562.

185. *Id.*

186. 17 U.S.C. § 301(a) "preempts 'all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106' when the work of authorship in which the rights are claimed falls 'within the subject matter of the copyright as specified by sections 102 and 103 of the copyright act.'" 775 F. Supp. at 563.

The district court found that ADAPTER, as a computer program, was copyrightable, and that CA's trade secret claims were in fact "equivalent" to, and therefore preempted by, its claims for infringement of its copyright. *Id.* at 565. To that court,

> [t]he alleged trade secret — ADAPTER — is the same entity in which CA alleges a copyright. While the generic elements of a trade secret and of a copyrightable work differ, the tort of misappropriation as particularly alleged and proved here and the infringement of CA's copyright boil down to the same thing — a right of action for the unauthorized reproduction of, and preparation of derivative works based on, ADAPTER. Merely because the elements of a "trade secret" and a "copyrightable work" are different does not avoid preemption.

Id. at 564.

Yet, the court suggested that if CA had made Arney a party and brought claims based on the "illegal *acquisition* of a trade secret, . . . the preemption analysis might be different, for there seems to be no corresponding right guaranteed to copyright owners by § 106 of the copyright act." *Id.* at 565.

In its June 22d opinion, the Second Circuit affirmed, finding that the district court had gone "beyond the pleadings and, in reaching its preemption decision, [taken] note of the case's factual environment" in isolating as the crux of both Computer Associates' trade secret and copyright infringement claims the allegation that Altai had copied ADAPTER into its ZEKE, ZACK and ZEBB programs. Computer Assocs. Int'l, Inc. v. Altai, Inc., Nos. 91-7893, 91-7935, 1992 U.S. App. LEXIS 14305, at *75 (2d Cir. June 22, 1992).

However, on December 17, 1992, after CA's petition for rehearing had directed the court's attention to "portions of the record below that were not included in the appendix for appeal," the Second Circuit withdrew its initial opinion. Leaving intact its discussion of copyright protection for non-literal elements of software, the court amended its analysis of the trade secret preemption issue.

The amended opinion vacated the district court's preemption ruling on CA's claims for wrongful acquisition of trade secrets and remanded those claims to the district court for a determination on the merits. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 716 (2d Cir. 1992). The district court had previously addressed the possibility that Altai had been on constructive notice

ses,[187] the court rejected them both and engaged in its own estimation, awarding CA $364,444 plus interest for damages resulting from the infringement of ADAPTER by OSCAR 3.4.[188]

CA and Altai each appealed, but Altai ultimately abandoned its appeal.[189] The Second Circuit thus faced only the two issues raised by CA on appeal.[190] CA challenged the district court's determination that OSCAR 3.5 had not infringed on CA's copyrights and disputed the district court's method for comparing OSCAR 3.5 with CA-SCHEDULER. CA also insisted that the district court erred in determining that CA's claims against Altai for misappropriation of trade secrets had been preempted by the federal copyright act.[191]

Although CA had copyrighted CA-SCHEDULER 2.1, which contained ADAPTER, the certificate of copyright registration identified this version as "derivative" of CA-SCHEDULER 1.0,

---

of Arney's breach of his confidentiality agreement with CA, which would have placed Altai under a duty of inquiry. Moreover,

> with regard to OSCAR 3.5, CA has a viable trade secret claim against Altai that must be considered by the district court on remand. This claim is grounded in Altai's alleged use of CA's trade secrets in the creation of OSCAR 3.5, while on *actual* notice of Arney's theft of trade secrets and incorporation of those secrets into OSCAR 3.4 . . . [w]here the use of copyrighted expression is simultaneously the violation of a duty of confidentiality established by state law, that extra element renders the state right qualitatively distinct from the federal right, thereby foreclosing preemption under § 301."

Id. at 719.

187. 775 F. Supp. at 567-71. 17 U.S.C. § 504 imposes copyright infringement liability in the amount of "the copyright owner's actual damages and any additional profits of the infringer, as provided by subsection (b)." Subsection (b), in turn, indicates that "in establishing the infringer's profits, the copyright owner is required to present proof only of the infringer's gross revenue, and the infringer is required to prove his or her deductible expenses and the elements of profit attributable to factors other than the copyrighted work." 775 F. Supp. at 567.

188. 775 F. Supp. at 572.

189. Altai had originally appealed on the grounds, *inter alia*, that it was not liable for the copying of ADAPTER in OSCAR 3.4. *Computer Assocs.*, 982 F.2d at 701.

190. *Id.*

191. *Id.*

which had never been registered.[192] Altai argued that the copyright thus extended only to the revisions of the original SCHEDULER program itself, and not to its subsequently-added subprogram, ADAPTER.[193] However, the district court concluded that, since "CA was the uncontested owner of the code [of ADAPTER] when CA registered its copyright in CA-SCHEDULER 2.1 as a derivative work, it was not required simultaneously to register separately every component part, such as ADAPTER, of that work. Otherwise complete copyright protection for a complicated program developed by the same author over a period of time would require dozens if not hundreds of registrations."[194]

Both the district court and the Second Circuit assumed that Altai had access to the ADAPTER code through Arney when it developed OSCAR 3.5.[195] Altai admitted copying ADAPTER into OSCAR 3.4, but not into OSCAR 3.5. Thus, to sustain its allegations of copyright infringement, CA was required to prove (1) that OSCAR 3.5 and ADAPTER were so "substantially similar" as to indicate that Altai had copied ADAPTER into OSCAR 3.5 and (2) that such copying involved copyrighted elements of ADAPTER.

B. *The District Court's Approach*

As the district court recognized, the nature of computer software fosters new interpretations of the basic principle that copyright protection applies not to ideas, but only to their expres-

192. 775 F. Supp. at 555.

193. *Id.* at 556.

194. *Id.*

195. The district court found that Altai had attempted "in good faith [to develop OSCAR 3.5 independently of Arney's pirated ADAPTER code] and adopted reasonable means to accomplish it," and in fact found that "neither Williams nor any of the other programmers who worked on OSCAR 3.5 took advantage of the ADAPTER code that was available to them, either by direct copying or by indirect copying from OSCAR 3.4." *Id.* at 558. Nonetheless, declining to resolve "the difficult factual issue that access presents in connection with OSCAR 3.5," the court assumed such access. *Id.* The Second Circuit adopted the same assumption "because we approve Judge Pratt's conclusions regarding substantial similarity." *Computer Assocs.*, 982 F.2d at 701.

sion.[196] "In the context of computer programs, many of the familiar tests for similarity prove to be inadequate, for they were developed historically in the context of artistic and literary, rather than utilitarian, works."[197] The court attacked *Whelan*'s "simplistic test for similarity between computer programs" as a glaring example of this "inadequacy."[198]

Noting the commentary of Professor Nimmer, but relying more heavily on the testimony of its own expert, Dr. Randall Davis of the Massachusetts Institute of Technology,[199] the district court found the *Whelan* analysis "fundamentally flawed."[200] The Third Circuit had erroneously assumed that each computer program contained only one "idea," and that the remainder of the program was protectable as a separate expression;[201] yet many complex programs contain various levels of subprograms, each of which may include its own unprotectable ideas.[202] Thus, *Whelan*'s narrow definition of "idea" would allow a developer to monopolize not only an arrangement of subprograms but also the subprograms themselves, even if these programs had approached the level of ideas by becoming "so standard or routine in the computer field as to be almost automatic statements or instruc-

196. 775 F. Supp. at 558 (citing Baker v. Selden, 101 U.S. 99 (1880); Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 42 (D. Mass. 1990); 17 U.S.C. § 102(b)).

197. 775 F. Supp. at 558.

198. *Id.*

199. The court appointed Dr. Davis in light of "the extensive technical evidence and expert testimony anticipated from both sides." *Id.* at 549.

Under Federal Rules of Evidence 706(a):

> The court may on its motion or on the motion of any party enter an order to show cause why expert witnesses should not be appointed, and may request the parties to submit nominations. The court may appoint any expert witnesses agreed upon by the parties, and may appoint expert witnesses of its own selection. . . . A witness so appointed shall advise the parties of the witness' findings, if any; the witness' deposition may be taken by any party; and the witness may be called to testify by the court or any party. The witness shall be subject to cross-examination by each party, including a party calling the witness.

FED. R. EVID. 706(a).

200. 775 F. Supp. at 560.

201. *Id.* at 559 (quoting 3 NIMMER & NIMMER, *supra* note 30, § 13.03[F], at 13-78.33).

202. *Id.* at 559.

tions written into a program."[203]

As Dr. Davis stressed, *Whelan* had also failed to distinguish between the "dynamic structure" of a program's "behavior," that is, the sequence in which the program performs its operations, and the "static" structure of the program's "text," or the order in which the written program provides for such operations.[204] In fact, there is no necessary connection between the two forms of structure: from a program's behavior "one can tell virtually nothing about its text."[205]

The court rejected not only the "intrinsic/extrinsic" test of *Arnstein* and *Krofft* but also *Whelan*'s "purpose/expression" test, in favor of an approach first advanced by Learned Hand in the context of literary works and recommended by Dr. Nimmer for application to software:

> Upon any work . . . a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the [work] is about and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the [author] could prevent the use of his "ideas" to which, apart from their expression, his property is never extended.[206]

The district court followed the "increasing generality" of the various levels of the software's design, "from object code, to source code, to parameter lists, to services required, to general outline."[207] The relative importance of these factors in the "sub-

---

203. *Id.*

204. *Id.* at 559-60.

205. *Id.* at 559. The district court suggested that the "behavior" of a computer program might qualify for patent protection instead of copyright protection, since 17 U.S.C. § 102(b) excludes "processes," "systems," and "methods of operation" from such protection. *Id.* at 560. However, the issue was moot "because CA's rights in this case are fully protected by viewing the ADAPTER program as text." *Id.*

206. *Id.* at 560 (quoting Nichols v. Universal Pictures, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931)). *See also* Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (2d Cir. 1936), *rev'g* 7 F. Supp. 837 (S.D.N.Y. 1934), *cert. denied*, 298 U.S. 669 (1936).

207. 775 F. Supp. at 561. The Second Circuit observed that "[w]hile the facts of a different case might require that a district court draw a more particu-

stantial similarity" analysis were rated by an expert as 1,000 ("code"), 100 (parameter lists), 100 (macros),[208] 1 (services) and 1 ("[general] organization chart").[209] The court found that there was no substantial similarity between the codes of ADAPTER and of OSCAR 3.5, since in creating OSCAR 3.5 Altai had replaced the ADAPTER code copied into OSCAR 3.4.[210] Although the two programs contained some similar macros and lists of parameters, the great majority of these resemblances were attributable to elements "in the public domain or dictated by the functional demands of the program."[211] In a direct renunciation of *Whelan*, the court minimized the importance of any similarities between the programs' organizational charts and their respective lists of services.[212] Therefore, the court held that "OSCAR 3.5 was not substantially similar to CA's ADAPTER program, that it was not copied from ADAPTER, and that it did not infringe CA's copyright on ADAPTER as contained in any of the registrations for CA-SCHEDULER."[213]

C. *The Second Circuit's Three-Part Test*

In affirming the district court's decision, the Second Circuit began from the "powerful" syllogism that "if the non-literal structures of literary works are protected by copyright; and if computer programs are literary works, as we are told by the legislature; then the non-literal structures of computer programs are protected by copyright."[214]

Like the district court, the Second Circuit found that *Whelan* displayed a "somewhat outdated appreciation of computer sci-

larized blueprint of a program's overall structure, this description is a workable one for the case at hand." Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 714 (2d Cir. 1992).

208. For definition of "macros," see *infra* note 321.

209. 775 F. Supp. at 562.

210. *Id.*

211. *Id.*

212. *Id.*

213. *Id.*

214. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 702 (2d Cir. 1992).

ence."[215] Using the lower court's approach as "a road map for our own,"[216] the Second Circuit enunciated a new three-step test, of "abstraction," "filtration" and "comparison," to determine the substantial similarity of the non-literal elements of different computer programs.

Step One: Abstraction

> [A] court would first break down the allegedly infringed program into its constituent structural parts.[217]
>
> . . . .
>
> [I]n a manner that resembles reverse engineering on a theoretical plane, a court should dissect the allegedly copied program's structure and isolate each level of abstraction contained within it. This process begins with the code and ends with an articulation of the program's ultimate function. Along the way, it is necessary essentially to retrace and map each of the designer's steps — in the opposite order in which they were taken during the program's creation.[218]

---

215. *Id.* at 706. In particular, "each subroutine is itself a program, and thus, may be said to have its own 'idea.' " *Id.* at 705.

216. *Id.* at 714.

217. *Id.* at 706.

218. *Id.* at 707. The court identified the steps in the development of a computer program (presented in condensed form below) as follows:

(1) The identification of a program's ultimate function or purpose.

(2) The breaking down or "decomposition" of the program's ultimate function into subtasks, which are also known as subroutines or modules.

(2a) If necessary, breaking down these subroutines further, into sub-subroutines.

(3) The arrangement of the subroutines or modules into organizational or flow charts, which map the interactions between modules that achieve the program's end goal.

(3A) In order to accomplish these interprogram interactions, a programmer must carefully design each module's parameter list, which tracks the form and substance of information passed between modules.

(3B) In fashioning the structure, a programmer will normally attempt to maximize the program's speed, efficiency, as well as simplicity for user operation, while taking into consideration certain externalities such as the memory constraints of the computer upon which the program will be run.

(4) Embodying the program's structure (the interrelationship of the modules and their functions) in a written language that the computer can understand, or "coding." This process involves:

(4.1) Transposing the program's structural blueprint into a

In this connection, the district court had identified levels of abstraction, in increasing order of generality, as: object code, source code, parameter lists, services required, and general outline;[219] it had proceeded to compare these corresponding elements of ADAPTER and OSCAR 3.5.[220] It had also found almost no copying of source code from ADAPTER to OSCAR 3.5.[221]

Step Two: Filtration

Then, by examining each of these parts [of the plaintiff's program] for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable material.[222]

"source code," written in a computer language such as COBOL, FORTRAN, BASIC or EDL; and then

(4.2) Translating or "compiling" the source code into "object code," which is the binary language composed of zeros and ones through which the computer directly receives its instructions.

(5) "Debugging" the program by running it on the computer in order to find and correct any logical or syntactical errors.

Id. at 697-98.

Cf. Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1229-31 (3d Cir. 1986) (identifying the stages of program creation, "moving from the general to the specific," as: (1) identification of the problem to be solved by the software; (2) outlining/flowcharting the solution, and the subroutines or modules that address specific aspects of the problems; (3) determining what data are needed, where along the program's operations the data should be introduced, how the data should be inputted, and how it should be combined with other data, and implementing data files to facilitate the handling of data produced by and used in the arrangement of subroutines and modules chosen; (4) coding the program into language understandable by the computer, first into (a) source code and then into (b) object code; (5) debugging the program; (6) documenting the program by producing written material to explain to the user how the program runs; and (7) maintenance of the program as the user develops new needs for the program), *cert. denied*, 479 U.S. 1031 (1987).

219. Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 560 (E.D.N.Y. 1991), *aff'd in part, vacated in part*, 982 F.2d 693 (2d Cir. 1992).

220. *Id.* at 560-62.

221. *Id.* at 562.

222. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 707 (2d Cir. 1992). The Second Circuit noted that the filtration procedure should be applied to the features of the allegedly *infringed* program — in this case, ADAPTER — before the comparison of this program's "kernel" to the allegedly *infringing* program. *Id.* This approach would preclude the consideration of irrelevant but copyrightable elements of the defendant's program; in addition, it would ensure

Besides "incorporated ideas," the "non-protectable material" removed at this stage of the analysis includes elements dictated by efficiency, elements dictated by external factors, and elements taken from the public domain. Each of these categories corresponds to an analogous doctrine in the literary copyright area.

The exclusion of elements dictated by efficiency parallels the copyright doctrine of merger, which allows the copying of the sole means of expressing an idea, on the grounds that "the idea and its expression are inseparable" in that instance.[223] Otherwise, the use of the idea would be restricted to the copyright holder. The most efficient programs, those that most closely embodied the idea or process behind the program's structure,[224] might be able to assume only a few variant forms.[225] Moreover, similarities between programs in this regard might be attributed to the industry-wide pursuit of efficiency rather than to copying.[226]

The elimination from the copyrightability analysis of elements dictated by external factors is analogous to the *"scenes a faire"* doctrine, by which copyright will not extend to " 'stock' or standard literary devices" necessarily linked to "a particular historical era or fictional theme."[227] The external factors giving rise to

---

that the comparison centered on the qualitative extent to which the defendant's program incorporated elements of the plaintiff's. *Id.* at 706.

Although the district court had instead subjected OSCAR 3.5 to the filtration analysis, the Second Circuit detected "no material impact on the outcome of this case. Since Judge Pratt determined that OSCAR effectively contained no protectable expression whatsoever, the most serious charge that can be levelled against him is that he was overly thorough in his examination." *Id.* at 714.

223. *Id.* at 707-08 (quoting Concrete Mach. Co. v. Classic Lawn Ornaments, Inc., 843 F.2d 600, 606 (1st Cir. 1988)).

Perhaps the prototypical application of the merger doctrine was Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 741-42 (9th Cir. 1971), in which the Ninth Circuit found inextricable the idea and expression of a jewel-encrusted pin in the shape of a bee. Given the extremely narrow range of types of such ornaments, the Court declined to extend copyright protection, which would amount to a monopoly, to one such version.

224. *Computer Assocs.*, 982 F.2d at 708.

225. *Id.*

226. *Id.*

227. *Id.* at 709 (quoting Hoehling v. Universal City Studios, Inc., 618 F.2d 972, 979 (2d Cir.), *cert. denied,* 449 U.S. 841 (1980)) (scenes involving German beer halls, German greetings such as "Heil Hitler" and the singing of certain

similarities among computer programs include:

> (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.[228]

German songs could appear in different portrayals of life in Nazi Germany without giving rise to copyright infringement). The court cited Q-Co Indus., Inc. v. Hoffman, 625 F. Supp. 608, 616 (S.D.N.Y. 1985), in which four program modules in a teleprompter program did not merit copyright protection because they would be essential to any such computer program. *Cf.* Burgess v. Chase-Riboud, 765 F. Supp. 233 (E.D. Pa. 1991) (play about Thomas Jefferson's alleged relationship with slave "concubine" infringed on novel that elaborated on slim historical account; many similarities between play and novel could not be traced to any historical account).

See also Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204, 209 (9th Cir. 1988) (similarities between two karate video games dictated by "constraints inherent in the sport of karate itself," including "[t]he number of combatants, the stance employed by the combatants, established and recognized moves and motions regularly employed in the sport of karate, the regulation of the match by at least one referee or judge, and the manner of scoring by points and half points. . . . Because of these constraints, karate is not susceptible of a wholly fanciful presentation.").

Whelan, citing this doctrine in support of its own holding that software components necessary to implement the program's purpose were not copyrightable, prefigured *Computer Associates'* removal of these elements from the copyrightability analysis. Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1236 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

228. *Computer Assocs.*, 982 F.2d at 909-10 (citing 3 NIMMER & NIMMER, *supra* note 30, at § 13.03[F][3], at 13-91, and Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256, 1262 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987)) (upholding denial of injunction against alleged infringer on grounds that externalities of cotton market led to similarities between programs). *See* Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204, 209 (9th Cir. 1988) (infringement claim rejected, in part because use of Commodore home computer imposed design constraints on karate game design); Manufacturers Technologies, Inc. v. CAMS, Inc., 706 F. Supp. 984, 995 (D. Conn. 1989) (since functioning of hardware package dictated type of navigational tools used in software's screen displays, copyright protection denied to that aspect of program). *See also* Softel, Inc. v. Dragon Medical & Scientific Communications, Inc., No. 87 Civ. 0167 (JMC) 1992 WL 168190 at 24-25 (S.D.N.Y. June 30, 1992) (citing second step of *Computer Associates* analysis in denying, on common use or efficiency grounds, infringement claim alleging substantial similarity due to use of hierarchy of menus, functional modules, external files and English language).

The Second Circuit exempted from copyright considerations material in the public domain, "by virtue of freely accessible program exchanges and the like"[229] or "if not standard, then commonplace in the computer software industry."[230]

The court observed that the district court had applied the filtration analysis to OSCAR 3.5 rather than to ADAPTER, thereby assuming the unnecessary duty of identifying protectable elements in the defendant's program that might not be found in the plaintiff's program and that would be irrelevant to the copyright infringement analysis.[231] This approach would also tend to minimize the effect of a "quantitatively small misappropriation which is, in reality, a qualitatively vital aspect of the plaintiff's protectable expression."[232] However, since the district court had found OSCAR 3.5 devoid of protectable expression, the results of its analysis would not deviate from those that would have been reached under the Second Circuit's test.[233]

The district court dismissed the majority of the similarities between ADAPTER's and OSCAR 3.5's parameter lists and macros as derived from the public domain or as dictated by the function of the program itself.[234] Analogously, the similarities in the lists of services required for the programs, and in the programs' respective organizational charts, were attributed to the specifications of the operating systems and applications with which ADAPTER and OSCAR 3.5 would be interacting.[235]

---

229. *Computer Assocs.*, 982 F.2d at 710.

230. *Id.* (citing Brown Bag Software v. Symantec Corp., 960 F.2d 1465, 1473 (9th Cir.), *cert. denied,* 113 S. Ct. 198 (1992) (quoting and affirming Telemarketing Resources v. Symantec Corp., 12 U.S.P.Q.2d (BNA) 1991, 1995-96 (N.D. Cal. 1989))).

231. *Id.* at 714.

232. *Id.*

233. *Id.*

234. *Id.* at 714-15.

235. *Id.* at 715. CA claimed that the organizational chart of ADAPTER was protectable even though the district court had found it "obvious" from the nature of the program's operation. Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 562 (E.D.N.Y. 1991). However, the Second Circuit held that the lower court had used this word to reflect that the organizational chart was dictated by the nature of the program itself; it accordingly held the organizational chart uncopyrightable under the *scenes a faire* doctrine. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d at 715.

Step Three: Comparison

Left with a kernel, or possibly kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program [to determine their substantial similarity].[236]

. . . .

At this point, the court's substantial similarity inquiry focuses on whether the defendant copied any aspect of this protected expression [*i.e.*, the "golden nugget" of the plaintiff's program], as well as an assessment of the copied portion's relative importance with respect to the plaintiff's overall program.[237]

The district court had at this stage identified, as the OSCAR 3.5 elements that were both copyrightable and were similar to those in ADAPTER, only a few of the parameter lists and macros; that court concluded that CA had not proved the "substantial similarity" of these elements to their counterparts in ADAPTER.[238] Reviewing the court's factual findings under the "clearly erroneous" standard, the Second Circuit found no error.[239]

IV. IMPLICATIONS OF *COMPUTER ASSOCIATES*

By its reliance on the testimony of a court-appointed expert, its emphasis on identifying the "golden nuggets" of copyrightable material in plaintiffs' programs, and its exclusion of efficiency-dictated elements from such copyrightable cores, the Second Circuit has undoubtedly encouraged competitors to duplicate features of best-selling computer programs. In this light, the court's

---

236. *Computer Assocs.*, 982 F.2d at 706.

237. *Id.* at 710.

238. 775 F. Supp. at 562.

239. *Computer Assocs.*, 982 F.2d at 715 (citing Federal Rule of Civil Procedure 52). Under Rule 52,

> [f]indings of fact shall not be set aside unless clearly erroneous, and due regard shall be given to the opportunity of the trial court to judge of the credibility of the witnesses. The findings of a master, to the extent that the court adopts them, shall be considered as the findings of the court.

FED. R. CIV. P. 52.

suggestion that software developers might be better served by patent than by copyright protection their programs seems worthy of consideration.

A. *Dangers of Expert Testimony*

As noted above,[240] the Second Circuit had previously limited the use of expert testimony in copyright cases to the initial determination of whether two works were so "substantially similar" as to indicate copying. Upon a finding of copying, only the trier of fact could evaluate whether the similarity was so great as to constitute copyright infringement.[241]

However, that precedent assumed that the triers of fact "comprise[d] the audience for whom the [works at issue were] composed."[242] As the court recognized in *Computer Associates*, "the highly complicated and technical subject matter at the heart of" computer software claims would probably be "somewhat impenetrable by lay observers — whether they be judges or juries — and thus, [would] seem to fall outside the category of works contemplated" by the precedent on expert testimony.[243] Thus, though maintaining the restrictions on the expert's role in "judging substantial similarity in copyright cases that involve the aesthetic arts, such as music, visual works or literature," the Second Circuit granted the district court the discretion to determine the use of expert testimony in software cases.[244] In this context, it noted that the district court, although depending heavily on the expert's report to unravel the technological complexities of the software infringement claim, had rightfully remained the trier of fact.[245]

Ironically, commentators have observed that the use of experts

240. *See* discussion of Arnstein v. Porter *supra* notes 39-42 and accompanying text.

241. *Computer Assocs.*, 982 F.2d at 713 (citing Arnstein v. Porter, 154 F.2d 464, 468 (2d Cir. 1946) and 3 NIMMER & NIMMER, *supra* note 30, § 13.03[E][3], at 13-78.12 (expert analysis appropriate in comparison of two works in their entirety)).

242. *Arnstein*, 154 F.2d at 473.

243. *Computer Assocs.*, 982 F.2d at 713.

244. *Id.*

245. *Id.*

can divert the court's focus from the direct comparison of the two programs at issue to "consideration of the behavior of the author of the accused work."[246] This detailed testimony, then, might be inappropriately applied to mask more equitable analyses.

The growing reliance of courts on their own appointed experts has also spawned procedural problems,[247] including the lack of a national referral system, the reluctance of potential witnesses to expose themselves to cross-examination, and special difficulties in discovery and *ex parte* communication.[248] Ironically, the in-court education of the judge may be hampered by the courtroom's own technological inadequacies.[249]

Even more dangerous are the substantive risks involved if the expert unduly influences the court, particularly with regard to interpretation of law.[250] In fact, at a recent symposium on computer law, Stephen D. Kahn, counsel for Computer Associates,

---

246. Conley & Peterson, *supra* note 132, at 468. By contrast, the expert's role in the determination of copyright infringement allegations involving software generally revolves around the comparison and evaluation of similarities in the programs' (1) names of variables and subroutines; (2) characteristics of variables, arrays, modules and interfaces; (3) sequences in lists, tests and calculations; (4) redundant code; (5) programmer's comments accompanying the computer code; (6) threshold values and default options; (7) internal representation of options (for example, by particular digits); (8) perpetuated errors; (9) output format; and (10) algorithms.
Id. at 453-68.

247. By contrast, "the patent code sensibly requires that the issue of obviousness be judged according to the opinion of an *ordinary* expert — that is, 'a person having ordinary skill in the art to which [the invention's] subject matter pertains' . . . . [and] wisely avoids reliance on the views of *expert* experts." John S. Wiley, Jr., *Copyright at the School of Patent*, 58 U. CHI. L. REV. 119, 144-45 (1991) (quoting 35 U.S.C. § 103).

248. Sheila L. Birnbaum & Gary E. Crawford, *Why Courts Hesitate To Appoint Experts*, NAT'L L.J., Oct. 26, 1992, at 16, 18.

249. Victoria Slind-Flor, *Tackling High Tech: Jurists Learn To Cope with the Brave New World*, NAT'L L.J., Oct. 19, 1992, at 1, 28 (quoting attorney observation that judges might request parties to prepare videotapes on the technical background of CD-ROM's, but that courtroom design, "where it's hard to find a blackboard or a grounded plug," does not support such information transfer).

250. *Cf.* Anthony L. Clapes et al., *Silicon Epics and Binary Bards*, 34 UCLA L. REV. 1493, 1574 (1987) ("Expert opinion is particularly helpful in differentiating independent development from copying in cases of comprehensive nonliteral similarity of computer programs.").

acknowledged that a full application of the first, or "abstraction," element of the Second Circuit's test could only practically be performed by software experts.[251] He suggested, though, that the *Computer Associates* district court had overly deferred to the court's expert, Dr. Randall Davis, who had not only testified for the copyright infringement plaintiff in *Gates Rubber Company v. Bando American, Inc.*[252] but had become a lecturer on the subject after the *Computer Associates* decision was issued.[253]

Specifically, Kahn faulted Dr. Davis for testifying that *Whelan* concerned a distinction between a program's "dynamic structure" (that is, the program's "behavior," or the sequence in which it performs operations) and protectable "static" structure (the arrangement of the program's "text" of coded instructions).[254] While agreeing that a program's "behavior" was not protectable by copyright, Kahn asserted that *Whelan* had focused neither on "behavior" nor on "static" structure in its Appendix A, a table comparing the sequence in which the order entry modules of the two laboratory management programs requested and manipulated data.[255] Elsewhere, Kahn has remarked of Dr. Davis, "I am not challenging his good will. . . . But buried in the 60-page report we got 48 hours before the trial, was a reference to *Whelan* as an 'infamous' decision. With hindsight, this tells me he had some views about *Whelan*."[256]

To reduce the potential for a tyranny of experts in the wake of

---

251. Transcript of seminar on *Computer Assoc. v. Altai*, 6th Annual Computer & Information Technology Law Institute, sponsored by The University of Texas School of Law and The Computer Law Association, Dallas, Texas, Sept. 17, 1992.

252. 798 F. Supp. 1499 (D. Colo. 1992), *modified*, No. 92-S-136, 1992 U.S. Dist. LEXIS 13601 (D. Colo. Aug. 12, 1992).

253. *Id.*

254. *See supra* note 204 and accompanying text.

255. Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1249 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987). The court reproduced this "comparison of the order entry modules [as] characteristic of [the expert's] comparisons of the other modules [which collectively were] probative, not dispositive of copyright infringements." *Id.* at 1248 n.46.

256. Victoria Slind-Flor, *Will Ruling Impact Borland's Fall Trial?*, NAT'L L.J., July 13, 1992, at 19, 26.

Altai's counsel, Susan Braden, acknowledged that she had been more familiar than Kahn with Dr. Davis' background before she suggested Davis' appoint-

Computer Associates, the court and the parties should carefully examine the expert and her potential biases, insist on a detailed and comprehensive report, and clarify the degree to which the expert's testimony relies on, reflects, or questions substantive law in the area.

B. *Effect on Software Developers*

The Second Circuit dismissed the arguments of Computer Associates that the new copyright restrictions of the abstraction/filtration/comparison test would discourage the development of software: "The 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."[257]

---

ment as the court's expert, but opined that Davis had not controlled the court. *Id.*

Cf. Slind-Flor, *Tackling High Tech*, *supra* note 249, at 28 (software litigation attorney quoted to effect that "[t]echnical experts are odd ducks. . . . They are not really controllable. You pay your money, and then they get up there on the witness stand and do what they want to do.").

257. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 711 (2d Cir. 1992). Indeed, the court found its approach compatible with that of *Feist Publications v. Rural Telephone Service Co., Inc.*, in which the Supreme Court had invalidated the notion that "copyright was a reward for the hard work that went into compiling facts." *Id.* (citing 111 S. Ct. 1282, 1291 (1991)). The Second Circuit saw this abandonment of copyright's "sweat of the brow" doctrine as weakening *Whelan*'s argument that copyright protection of non-literal elements would "provide the proper incentive for programmers by protecting their most valuable efforts." *Id.* (quoting *Whelan*, 797 F.2d at 1237). However, *Computer Associates* did not mention that *Whelan* had itself refused to equate exertion with copyrightability:

> [F]act intensive works are given . . . limited copyright coverage [because] there are only a limited number of ways to express factual material. . . .
>
> The fact that it will take a great deal of effort to copy a copyrighted work does not mean that the copier is not a copyright infringer. The issue in a copyrighted case is simply whether the copyright holder's expression has been copied, not how difficult it was to do the copying. Whether an alleged infringer spent significant time and effort to copy an original work is therefore irrelevant to whether he has pirated the expression of an original work.

797 F.2d at 1236-37. *See supra* note 105 and accompanying text.

Foreshadowing *Computer Associates*, the Third Circuit added, "[e]ven if the

In this vein, commentators have generally predicted that the "restrictive multi-step test" of *Computer Associates* will inhibit software market leaders from bringing copyright infringement actions, since the court would then define in detail those elements of their programs that *would* be permissible targets of copying.[258]

However, competitors could profitably apply the lessons of *Computer Associates* even without recourse to a court-commissioned copyrightability map. Guided by their own legal and computer consultants, aggressive developers could dissect, for instance, all leading spreadsheet models to map their similarities at all levels of abstraction, from object code through source code to overall structure. After identifying the common features that might qualify as "ideas," those that might be dictated by the nature of the program or the hardware, and those that seem to be dictated by efficiency concerns, developers could then choose the level of infringement risk they would assume by selectively incorporating more or fewer of these features into their programs.[259]

---

product of the alleged infringer's efforts is a mixture of old and new elements, that would not protect it from the charge of infringement." *Id.* at 1237 n.32.

258. Richard Raysman & Peter Brown, *Computer Law: Copyright Protection for Software Redefined*, N.Y. L.J., July 14, 1992, at 3, 7. *Computer Associates* has generally been characterized as "a blow to major computer companies seeking to protect best-selling programs" because it indicates that "competitors may be able to copy important elements of those programs and thereby grab market share." Jonathan S. Moses, *Court Eases Copyright Rules on Software*, WALL ST. J., June 24, 1992, at B1.

See also Lee Gesmer, *Decisions May Signal a Judicial Turnabout*, NAT'L L. J., Jan. 18, 1993, at S2, S4 ("It is likely . . . that the flood of software copyright infringement cases triggered by *Whelan* will slow to a trickle as plaintiffs encounter the challenges created by *Computer Associates*."); Richard Brandt et al., *Bit by Bit, Software Protection Is Eroding*, BUS. WK., July 20, 1992, at 86-87 ("[C]ompetition in this industry will probably increase as entrepreneurs feel legally safer. It could also reduce the number of legal victories awarded to market leaders [such as Lotus] who now rush to court when rival programs seem to imitate their work.").

259. The *Computer Associates* analysis should throw a new light on the protection of "object-oriented" programs. *See, e.g.*, John W. Verity & Evan I. Schwartz, *Software Made Simple: Will Object-Oriented Programming Transform the Computer Industry?*, BUS. WK, Sept. 30, 1991, at 92 [hereinafter *Software Made Simple*]. The major advantage of this new trend in technology

> is the ability to build large programs from lots of small, prefabricated ones. That's possible because objects completely change the traditional relationship between programs and data, which have been strictly segregated for 40 years. As the old term "data processing"

C. *Appropriateness of Software Copyrights*

1. *Copyright v. Patent*

Computer Associates observed that "the indiscriminating availability" of copyright registration may not be appropriate for "the highly dynamic technology of computer science," since it forces courts to "attempt to fit the proverbial square peg in a round hole."[260]

As one commentator has noted, "[t]he traditional distinction

---

implies, programs ordinarily act on data — simple lists of numbers or customer names, for example. An object, in contrast, encapsulates programs and data in one self-contained unit, which fully describes some real-world entity. . . .

This simple idea provides tremendous benefits. Software objects can be built to represent just about everything — from an abstract concept, such as an insurance policy, to a specific thing or person, such as Duke Ellington, American composer and musician, 1899-1974. More important, objects can be created that perform certain common tasks — sorting, for example. Once perfected, such objects are infinitely reusable, so programmers don't have to reinvent the wheel every time.

Id. at 94.

It remains to be seen how an object-oriented program would fare under the *Whelan* test if it served the same "purpose or function" as another such program or as less advanced software.

In either of the situations, the more comprehensive analysis of *Computer Associates* would examine the various structural levels of the plaintiff's program. The "abstraction" analysis of an "object-oriented" program would encompass not only its general function and the manner in which it had been assembled from "objects," but also the structure of the "object" subprograms themselves.

Under a broad interpretation of the "filtration" step, the structure of the program and of the "objects" would be uncopyrightable if the court found that their design had been dictated by efficiency. Moreover, program elements that are in the public domain, or are necessary for compatibility with the specific hardware and software used would also remain subject to copying by competitors. *See Software Made Simple*, *supra*, at 98-100 (Object Management Group, including more than 160 computer and software makers and customers, is currently attempting to develop electronic system to distribute software objects across networks involving many different types of computers; Digital Equipment, Sun and Hewlett-Packard are collaborating to produce necessary software). An "object's" original arrangement of information might qualify for copyright protection; finally, the information contained in each object might be copyrightable in its own right.

The "comparison" test would match the copyrightable features of the "object-oriented" program to the features of an allegedly infringing program.

260. *Computer Assocs.*, 982 F.2d at 712.

between the two domains is that patent excludes unauthorized practice of an invention, whereas copyright excludes unauthorized copying of a writing. For the most part, the demarcation between these two domains is reasonably clear. But a problem is posed by the computer program, because it appears that in this area uniquely, the medium is the message."[261]

The Patent Act of 1952, as amended,[262] protects "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof,"[263] so long as the work is both "novel"[264] and "non-obvious."[265] However, "the issue of statutory subject matter is still largely undefined and in a state of constant flux."[266] The two-step test generally employed, while roughly analogous to copyright law's "idea versus expression" distinction, does afford protection for certain ideas, or "mathematical algorithms," that are applied to natural processes:

> First, does the claim directly or indirectly recite a mathematical algorithm? If not it is not barred by § 101. But if the claim recites a mathematical algorithm, step two must be addressed: Is the mathematical algorithm either (a) implemented in a specific manner so as to define structural relationships between physical elements of the claim (apparatus) or to refine or limit certain steps (process), the claim being otherwise statutory [*i.e.*, patentable], or (b) applied to process steps themselves part of an overall statutory process? If so, the claim is statutory; if not, it may be non-statutory. In the application of the second step, the claim should be given its broadest reasonable interpretation consistent with the specification. Moreover, the claim must be analyzed as a whole; dissection such that the mathematical algorithm is considered prior art, with the remainder of the claim tested under § 102, is improper.[267]

---

261. COMPUTER LAW, *supra* note 72, § 3A.09, at 3A-86.2. Bender notes that "there is no software case discussing the potential conflict [between copyright and patent protection]. [I]n previous instances of conflict the patent has proven to be a survivor." *Id.* at 3A-87.

262. 35 U.S.C. §§ 100-376 (1988).

263. *Id.* § 101.

264. *Id.* § 102.

265. *Id.* § 103.

266. COMPUTER LAW, *supra* note 72, § 3A.01, at 3A-2.3.

267. *Id.* at 3A-2.3 to 3A-2.4. Bender indicates that:

These determinations remain fact-sensitive despite several Supreme Court decisions in the area.[268] Indeed, "the lower courts have been comparatively silent on the issue of statutory

---

> [N]o case appears to define just what constitutes a mathematical algorithm. Rather, the cases tend, without discussion, to denominate each algorithm in question as mathematical or non-mathematical. Accordingly, the counselor is left to the process of induction in an attempt to discern one from the other. In the context in which the claim language arose, courts have held as follows.
>
> "Calculating" a difference recites a mathematical algorithm. "Summing" electrical signals to form another electrical signal appears to recite a mathematical algorithm . . . but a claim reciting a mathematical algorithm which converts electrical signals into another form is statutory. This last conclusion holds even though the physical apparitions comprising the electrical signals can be expressed in mathematical terms. Finally, steps such as "executing," "compiling," "storing" and "examining" formulas do not directly or indirectly recite mathematical algorithms, mathematical formulas, or calculations.

Id. at 3A-3.

See also Walter A. Effross, *Software Related Patent Is OK'd*, COMPUTER LAW STRATEGIST, Apr. 1992, at 1 (analyzing Arrhythmia Research Technology v. Corazonix, 958 F.2d 1053 (D.C. Cir. 1992), in which the court extended scope of patentable subject matter to include process and apparatus for electronically analyzing cardiograph signals).

268. Although the Supreme Court has not addressed the copyright protection of computer software programs, it has recognized the patentability of software-related inventions. COMPUTER LAW, *supra* note 72, § 3A.11, at 3A-95 (inclusion of algorithm and programmed digital computer did not render process for molding raw synthetic rubber nonstatutory, since these elements were applied in structure or process which, when considered as a whole, is performing a function which patent laws are designed to protect, such as transforming or reducing an article to a different state or thing) (citing Diamond v. Diehr, 450 U.S. 175 (1981)). *Cf.* Parker v. Flook, 437 U.S. 584 (1978) (denying patent protection to method for calculating alarm limits in catalytic chemical conversion of hydrocarbons, because algorithm the only novel aspect of method, because remainder of method poorly described, and because claim would broadly preempt similar uses of algorithm); Gottschalk v. Benson, 409 U.S. 63 (1972) (general method of conversion of data from binary-coded-decimal form to binary form not patentable, since conjoined neither to apparatus nor to specific application, and since patent protection would preempt underlying mathematical formula).

The Patent Office's guidelines on evaluation of claims including mathematical algorithms and software do not automatically deny patent protection to processes or inventions including such elements. COMPUTER LAW, *supra* note 72, § 3A.04, at 3A-50.16 (discussing *"Patentable Subject Matter — Mathematical Algorithms or Computer Programs,"* Manual of Patent Examining Procedures § 2110 (1981)).

subject matter in the computer program context."[269]

The relative advantages of patent protection over copyright protection in the software context[270] are: the ability to exclude others from employing the software's algorithms, as opposed to prohibiting merely the expression of the program's underlying "idea"; the ability to prosecute those who have independently created the program; the absence of any requirement that the plaintiff prove copying of its work by the defendant; the clearer identification of the protected elements of the program;[271] and the lack of a requirement that these elements be treated by the plaintiff as trade secrets. In addition, patent holds the promise of protecting in one integrated form the entire "business methods" involving computer programs.[272]

The disadvantages of this form of protection include: the uncertainty regarding the scope of the Patent Act's § 101 as applied to software;[273] a patent's cost to obtain and maintain; its inappli-

269. COMPUTER LAW, *supra* note 72, § 3A.03[2], at 3A-46.

270. *Id.* § 3A.10, at 3A-88 to 3A-89.

271. *Cf.* Randall M. Whitmeyer, *A Plea for Due Process: Defining the Scope of Patent Protection for Computer Software*, 85 NW. U. L. REV. 1103, 1136 (1991) (despite apparent advantages in proving patent infringement, copyright infringement may be less difficult to prove than at first appears: element of copying is often easy to prove because of program's widespread availability or parties' previous contractual relationship, and programmers can place idiosyncratic instructions, or "fingerprints," in their programs to reveal copying).

272. COMPUTER LAW, *supra* note 72, § 3A.07, at 3A-68.1 to 3A-68.2 (discussing patents for various computer-related apparatus for industrial use. Portions of the relevant patent documents are reproduced as appendices to Bender's treatise).

Bender also notes industry rumors that the developers of "a highly successful spreadsheet program" had been advised during its development that it could not be patented; suggesting that "today such a patent application might well be held to claim appropriate subject matter, and issue as a patent," he raises the specter of monopolization of entire new fields of software by those "who think they've got the next spread sheet" and are obtaining patents on their software. *Id.* § 3A-11, at 3A-96 to 3A-97.

273. Two recent decisions by the Board of Patent Appeals and Interferences have raised new questions regarding the scope of § 101 of the patent law. *See* E. Robert Yoches, *Once More into the Breach: The U.S. Patent and Trademark Office Renews Its Attack on Subject Matter Patentability of Computer Programs*, 15 COMPUTER REP. 863 (1992) (analyzing *Ex Parte* Akamatsu, Appeal No. 91-3230 (Mar. 20, 1992) and *Ex Parte* Alappat, Appeal No. 91-12778 (Apr. 22, 1992), both of which strictly construe § 101 in the context of "means-plus-function" elements).

cability to data bases[274] and to documentation; the enforceability of patent claims only against conduct that occurs after the issuance of a patent, a process which itself may take two years; its limited duration of seventeen years;[275] the unsettled status of the availability of injunctive relief under patent law; the ability of competitors to obtain the patent itself as a public document; and the implementation in antitrust suits of consent decrees setting limits on the royalties at which the parties can license their patents.[276]

---

274. However, *Feist* appears to have limited significantly the range of databases that can be copyrighted. *See supra* note 129 and accompanying text.

Generally, a database qualifies for protection under the Copyright Act as a "compilation," or "[a] work formed by the collection and assembling of preexisting materials or of data that are *selected, coordinated, or arranged* in such a way that the resulting work as a whole constitutes an original work of authorship." 17 U.S.C. § 101 (emphasis added). However, the Act limits the copyright compilation to "the material contributed by the author of such work, as distinguished from the preexisting material employed in the work." 17 U.S.C. § 103(b).

Under this analysis, *Feist* declined to protect as a compilation a telephone directory's "white pages," which listed names, addresses, and telephone numbers. When determining whether a fact-based work is an original work of authorship rather than a mere collection of facts, the Court held, the primary emphasis should be on the "selection, coordination, or arrangement" of the facts. Feist Publications, Inc. v. Rural Tel. Serv. Co., 111 S. Ct. 1282, 1293 (1991).

275. COMPUTER LAW, *supra* note 72, § 3A.10, at 3A-89 (citing 35 U.S.C. § 154). However, "[c]opyright has a term of protection which, in the software milieu, can only be called extraordinary — generally the author's life plus fifty years, or (for anonymous works, pseudonymous works, and works made for hire) seventy-five years." *Id.* § 4.02, at 4-4.12 (citing 17 U.S.C. §§ 302-305).

Moreover, there are indications that the effective term of patent protection provides more than adequate safeguarding of some forms of computer-related technology. *See, e.g.,* Edmund Andrews, *Rich in the 90's on Ideas Hatched in the 50's*, N.Y. TIMES, Nov. 13, 1992, at A1, (profiling inventor who was paid a total of more than $100 million by "more than a dozen big companies" in previous year on patents "based on inventions dating back to the 1950's, for automated manufacturing systems that use robots and computerized machines that can see"; indicating that inventor subdivided some of his earliest patents and nursed them through Patent Office so that some of his decades-old inventions did not receive patents until previous year); Edmund Andrews, *There's Cash in Mining the Courts*, N.Y. TIMES, Nov. 9, 1992, at D2 (Texas Instruments amassed hundreds of millions of dollars more in royalties in the last year or two by demanding licenses to old patents on computer chip technology than it made from selling products).

276. COMPUTER LAW, *supra* note 72, § 3A.10[2].

While it remains an open question whether these drawbacks outweigh the advantages of software patents,[277] an increase in patent filings by software developers is expected in response to the current uncertainties regarding copyright protection.[278]

2. *Report of the Office of Technology Assessment*

Noting that courts and a commentator had proposed that patent registration might be more suitable than copyright for protecting intellectual rights in software, the Second Circuit observed that "the resolution of this specific issue could benefit from further legislative investigation — perhaps a CONTU II."[279] While such a commission has not yet been convened, in

---

277. The length of the patent prosecution period is generally not a major impediment (and is becoming even less troublesome). Moreover, the commercial lifetime of the invention embodied in, or in the application of, the software, if worthy of a patent, is likely to be at least several years. Enforcement of patents has become less difficult. And the perception that source (or even object) code need be disclosed and published to secure the patent is generally just plain incorrect.

Id. § 3A-11, at 3A-96 (footnote omitted).

By contrast, another leading commentator has eliminated patent law as a viable option for the protection of most application program code. Most programs, though products of significant efforts to define, outline, and implement a method for performing tasks on a computer, simply do not manifest sufficient novelty or non-obviousness to merit patent protection. Even for those programs that are novel and non-obvious, the time and cost of obtaining protection may not be worth the effort, particularly if the product is not expected to have a long life cycle. Furthermore, to obtain a patent, the inventor must disclose the art to the Patent Office and ultimately to the public. The inventor might very well wish not to do this, for such disclosures would facilitate access by others to the inner workings of the patentee's invention and would destroy any trade secrecy claim.

Peter Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1076 (1989) (footnotes omitted).

278. Brandt, *supra* note 256, at 88.

279. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 712 (2d Cir. 1992) (citing Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 560 (E.D.N.Y. 1991); Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78, 91 (D. Mass. 1992); Randall M. Whitmeyer, *A Plea for Due Processes: Defining the Proper Scope of Patent Protection for Computer Software*, 85 NW. U. L. REV. 1103, 1123-25 (1991)).

The CONTU Report itself suggested:

Any legislation enacted as a result of these recommendations should be subject to periodic review to determine its adequacy in the light of

May 1992 the Congressional Office of Technology Assessment (OTA) released a 228-page report entitled *Finding A Balance: Computer Software, Intellectual Property and the Challenge of Technological Change*.[280]

In reviewing this area, the OTA identified "three principal policy areas that Congress may wish to address": the appropriate scope of copyright protection for software, the scope of patent protection for software-related inventions, and the applicability of copyright law to users of digital information.[281] While the third of these topics is beyond the scope of this article, the OTA's treatment of the first two issues reveals the degree to which both copyright and patent protection remains unresolved in the software context. Although the OTA did not make definitive recommendations, the sweeping nature of the options considered indicates that these areas may merit radical change.[282]

> continuing technological change. This review should especially consider the impact of such legislation on competition and consumer prices in the computer and information industries and the effect on cultural values of including computer programs within the ambit of copyright.

CONTU REPORT, *supra* note 12, at 3.

280. *See* FINDING A BALANCE, *supra* note 3. Among the nineteen members of the OTA's Advisory Panel for this investigation was none other than the ubiquitous Dr. Randall Davis, the court's expert in *Computer Associates* and the plaintiff's expert in Gates Rubber Co. v. Bando Am., Inc., 798 F. Supp. 1499 (D. Colo. 1992). *See infra* note 300 and accompanying text.

281. FINDING A BALANCE, *supra* note 3, at 28.

282. The Office of Technology Assessment noted that, with respect to each of these issues, Congress faced the questions of whether to act at all, when to act, and how comprehensively to act. *Id.* at 28-29. In lieu of introducing complementary or *sui generis* patent and/or copyright regimes tailored to computer software, Congress might elect to "explicitly affirm the status quo," to "make small adjustments at the margins of copyright and patent (*e.g.*, through procedural changes)" or "clarify or modify the scope of patent and/or copyright (*e.g.*, through definitional changes), but leave the basic paradigms the same." *Id.* at 29.

While Congress could avoid immediate action and instead continue to survey the developing law and technology,

> incremental accommodations through the case law may conflict over time, as the case law continues to evolve. As is the case with current legal uncertainties, the uncertainties that ensue will affect smaller/poorer firms and individuals (that do not have the resources to "ride it out") more than large firms with deep pockets.

Id.

Among the alternative proposals to clarify the protection of software are:

> 1. Revising the Copyright Act's Section 102, which specifies the "subject matter of copyright," explicitly to include or to exclude computer languages, algorithms, design specifications, user and other interfaces, and/or other aspects of software;
>
> 2. Creating a special category for computer programs in the Copyright Act, rather than treating them as "literary works";
>
> 3. Confining the treatment of software as a "literary work" to the code as text, rather than including the program's operational behavior or its user interfaces, which could be addressed by new statutory provisions;
>
> 4. Removing computer programs entirely from the scope of copyright protection, and protecting them instead "under a *sui generis* framework, including protection for the program code, as well as other elements of program functionality and design."[283]

Similarly, the report suggested that Congress "[r]efine the definition of patentable subject matter to provide guidance to the courts and [Patent Office]," particularly with regard to whether processes incorporated in software or mathematical algorithms are patentable; alternatively, it proposed the creation of a special patent framework, characterized by "a shorter term, lower criteria for inventiveness, and/or special exemptions for infringement" for "software-related inventions and/or algorithms."[284]

V. PROGENY OF *COMPUTER ASSOCIATES*

In the first seven months since its original release, the *Computer Associates* test was cited by several courts in varying contexts. One decision emphasized the test's elimination of efficiency-dictated elements from the copyrightability analysis;[285]

---

283. *Id.* at 30-32.

284. *Id.* at 33.

285. In Softel v. Dragon Medical & Scientific Communications, Inc., No. CIV. 0167 (JMC), 1992 WL 168190 (S.D.N.Y. June 30, 1992), the plaintiff sought to protect as copyrightable:

(1) the mere use of hierarchical menus, which, the court noted, were "one of the most efficient and user-friendly interfaces." *Id.* at *24;

(2) the use of functional modules, although "the evidence established that

another found it inapplicable to a situation in which the defendant admitted taking, reproducing and using without modification copies of the plaintiff's program.[286] Although *Computer Associates* had specifically declined to address whether screen displays were copyrightable as "audiovisual works,"[287] the Second Circuit's rejection of *Whelan* was cited as support for denying copyright protection to a user interface.[288] However, another court has found a plaintiff's file structures, screens and reports, and

---

programmers commonly write code in modules which perform each necessary subtask" and though this feature, in any case, would require the additional factor of the modules' interrelationships to qualify as a structural element." *Id.*;

(3) the use of external files, which "are commonly used in interactive computer programs because of the extensive amount of information which must be organized." *Id.*; and

(4) the use of English language commands, which use was "logical, pervasive, and the most effective way that the programmer can keep track of the available commands." *Id.* at * 25.

However, the district court applied the *Computer Associates* test to filter out, as "merely stock elements which are commonly used or mandated by efficiency considerations," all of the elements of medical diagnosis programs that the plaintiff had asserted were copyrightable, and thus was not required to perform the third step of the test. *Id.* The court emphasized that "after *Altai* it is now settled law in this Circuit that expression which is standard or the most efficient means of accomplishing a task merges with the idea and is not entitled to copyright protection." *Id.*

286. Data Gen. Corp. v. Grumman Sys. Support Corp., 803 F. Supp. 487 (D. Mass. 1992).

287. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 703 (2d Cir. 1992).

288. Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006 (N.D. Cal. 1992) refused to protect the cumulative "look and feel" of the Apple Macintosh's graphical user interface, the individual features of which were duplicated by rival programs, Microsoft's Windows 3.0 and Hewlett-Packard's NewWave, in their operation on non-Macintosh computers.

However, the five similarities claimed by Apple were each subject to a limiting doctrine, such as *scenes a faire* or merger, which would render them unprotectable "ideas":

(1) use of windows to display multiple images on a computer screen and facilitate interaction with the information contained in the windows;

(2) use of icons to represent familiar objects from the office environment and facilitate organization of information stored in the computer's memory;

(3) manipulation of icons to convey instructions and to control operation of the computer;

(4) use of menus to store information or functions of the com-

transaction codes copyrightable under the "abstraction-filtration-comparison" analysis.[289]

---

puters in a place that is convenient to reach, but saves screen space for other images; and

(5) opening and closing of objects as a means of retrieving, transferring or storing information.

Id. at 1026.

Citing *Synercom's* comparison of input formats to the stick-shift arrangement of an automobile, the Second Circuit analogized the visual displays and user commands of the Macintosh's user interface to the car's "dashboard, steering wheel, gear shift, brakes, clutch and accelerator." *Id.* at 1023 (citing 462 F. Supp. at 1013). Their unusual metaphoric nature, the court concluded, would not overcome the interface's "purely functional" status, and thus its noncopyrightability. *Id.*

The court agreed with *Computer Associates* that *Whelan* overly favored market leaders at the expense of those who were "stifl[ed] from improving or extending that expression." *Id.* at 1025. It observed that "some visual displays are or become so closely tied to the functional purpose of the article that they become standard," as evidenced by the "almost invariable incorporation of [features of the Macintosh's graphical user interfaces] in most graphical user interfaces." *Id.* at 1023. To allow Apple to copyright "for decades," *id.* at 1025, the interfaces in Windows and NewWave would "produce its own negative effects by inhibiting the adoption of compatible standards [which] would enlarge the market for computers by making it easier to learn how to use them." *Id.*

289. CMAX/Cleveland, Inc. v. UCR, Inc., 804 F. Supp. 337, 352-53 (M.D. Ga. 1992), rejected *Whelan's* idea/distinction test in favor of the *Computer Associates* three-step test. In determining that the defendant had infringed plaintiff's software, which was designed to aid the management of "rent-to-own" furniture and appliances, the court found the following non-literal elements of the plaintiff's program copyrightable:

(1) "file structures," that is, the selection and arrangement of field definitions within a file. *Id.* at 354 (citing Eckes v. Card Prices Update, 736 F.2d 859, 862-63 (2d Cir. 1984) (extending copyright protection to subjective selection and arrangement of items)). Since these structures were "not alphabetic or otherwise systematic, [or] functionally significant in any respect," they were not found to be dictated by the industry. *Id.*

(2) screens and reports, under the same analysis as above. *Id.* at 355; and

(3) transaction codes, which

> tell a computer how to act on a transaction. The idea behind a transaction code is that it is an abbreviated representation of a shortcut to the execution of a particular transaction. Plaintiff's expert testified that transaction codes were a major part of program design, that they could be anything, and that they were unique to the particular system designed. Defendants' expert did not testify that these codes were dictated by efficiency or by the industry. Rather, he testified that it would not make sense to change the codes because then the employees who were familiar with the [plaintiff's] transaction codes would have to learn new ones. This is not an external factor, which would negate

Of particular interest are two district court decisions construing the *Computer Associates* test. In Colorado, the Second Circuit's analysis has been used in conjunction with that of *Whelan*; in Massachusetts, the three-step test has been compared favorably to the district court's own.

Finally, recent decisions by the Circuit Courts of Appeals for the Federal Circuit and the Ninth Circuit have cited *Computer Associates* in legitimizing the practice of reverse engineering, a technological practice which itself may generate even more occasions for the application of the "abstraction/filtration/comparison" test.

A. *Gates Rubber Co. v. Bando American, Inc.*[290]

The district court for the District of Colorado employed the *Computer Associates* test as a fail-safe, to remove from the copyrightability analysis any program elements that are "ideas" and would survive *Whelan*. Though counteracting *Whelan*'s restriction of a program's "idea" to its purpose, the *Gates Rubber* test perpetuated the Third Circuit's conflation of expert and lay testimony.

Gates Rubber Company, the leading manufacturer of industrial belts, alleged that competitor Bando American's program, Chauffeur, had infringed Gates' copyright in Design Flex 4.0, a program used by salesmen in the field to determine the appropriate replacement belts for customers. The programs employed identical mathematical formulas and constants in designing drives and determining belt size.[291]

After a hearing limited to the appropriateness of awarding injunctive relief to Gates, the court found "more than ample evidence to support the inference that [Bando] had access to the Design Flex program."[292] Moving to the second prong of the copyright infringement test, the inquiry regarding whether the

---

the copyrightability of the transaction codes. Consequently, the transaction codes are protected.

Id.

290. 798 F. Supp. 1499 (D. Colo. 1992), *modified*, No. 92-S-136, 1992 U.S. Dist. LEXIS 13601 (D. Colo. Aug. 12, 1992).

291. *Id.* at 1503.

292. *Id.* at 1509.

two programs were substantially similar, the court found it impractical to attempt to categorize the various tests advanced by earlier decisions, which had been decided "on a case-by-case basis [that] led to a proliferation of different descriptive terms associated with particular holdings which have generated more heat than light."[293]

The court identified three different tests for the determination of substantial similarity. The first of these was *Whelan's* collapsing of the extrinsic/intrinsic test into a single inquiry incorporating both lay and expert opinions. Second, the court characterized *Computer Associates'* "abstraction test" as a "rather extreme alternative [which] may be rendered unimportant if [it] is used as a prelude to, instead of as a substitution for" the *Whelan* test.[294] Finally, the court cited *Dawson v. Hinshaw Music, Inc.*,[295] in which the Fourth Circuit, in evaluating a copyright claim involving a musical arrangement, had modified the intrinsic test to allow the substitution in some circumstances of the impressions of an ordinary lay observer for those of a specialized audience.[296]

---

293. *Id.* at 1510-11.

294. *Id.* at 1513. Certainly, *Whelan*'s expansive definition of protectable "expression" would add considerable weight to the "golden nugget" isolated by *Computer Associates*. For example, as discussed further in the context of Softel v. Dragon Medical & Scientific Communications, Inc., No. CIV. 0167 (JMC), 1992 WL 168190 (S.D.N.Y. June 30, 1992), discussed *supra* note 285, elements dictated by efficiency might be excluded from the "substantial similarity" analysis by *Computer Associates* but not necessarily by *Whelan*.

295. 905 F.2d 731 (4th Cir.), *cert. denied*, 111 S. Ct. 511 (1990).

296. When conducting the second prong of the substantial similarity inquiry, a district court must consider the nature of the intended audience of the plaintiff's work. If, as will most often be the case, the lay public fairly represents the intended audience, the court should apply the lay observer formulation of the ordinary observer test. However, if the intended audience is more narrow in that it possesses specialized expertise, relevant to the purchasing decision, that lay people would lack, the court's inquiry should focus on whether a member of the intended audience would find the two works to be substantially similar. Such an inquiry may include, and no doubt in many cases will require, admission of testimony from members of the intended audience or, possibly, from those who possess expertise with reference to the tastes and perceptions of the intended audience.

Id. at 736.

Dawson had agreed with *Whelan* that "only a reckless indifference to common sense would lead a court to embrace a doctrine that requires a copyright

From these variant tests, the *Gates Rubber* court derived its own: it applied the intrinsic/extrinsic test of *Whelan*, giving "substantially greater weight to the extrinsic prong," which nonetheless was "limited dramatically."[297] As had *Dawson*, the court acknowledged the "importance and technical nature of the expert testimony,"[298] but restricted the impact of that testimony to factual determinations. Once these expert opinions were received, the court proceeded to apply the *Computer Associates* abstraction/filtration/comparison test, "to ensure that [the previous tests did] not allow for the protection of any unprotectable ideas."[299]

Under the extrinsic test, the court heard the testimony of three experts: Dr. Randall Davis, who had been the court's expert in *Computer Associates*, here testified for the plaintiff;[300] Dr. William Dorn, the defendants' expert; and Dr. William Clancey, who had been selected by Drs. Davis and Dorn as an independ-

---

case to turn on the opinion of someone who is ignorant of the relevant differences and similarities between two works." *Id.* at 735.

297. *Gates Rubber Co.*, 798 F. Supp. at 1513-14.

298. *Id.*

299. *Id.* at 1514.

> This Court is of the opinion that it is far preferable, especially in an area of legal and technological sophistication as complex as this area of copyright protection, to draw upon a larger arsenal of facts in order to design or derive the appropriate legally significant facts. Once these are gathered and expert opinion is heard, the court can then analyze which portions of the program, according to the expert testimony, infringes [sic] the protected expression.

Id. at 1511. Thus, the court found it "only logical that [the *Computer Associates*] analysis would be undertaken after the application of the substantial similarity test, at which time the fact-finder has determined whether the work is a 'plagiarizing work' of the copyrighted work." *Id.* at 1516. Rejecting Gates's argument that the "filtration" process should precede the substantial similarity analysis, the court indicated that tests performed in that order could "eviscerate the application of" the *Whelan* test, without adding any compensating advantages. *Id.* Moreover, the reservation of the *Computer Associates* test for the final stage of the process "serves as a guard against unprotectable elements being considered in the legal conclusion of whether there is infringement (based on the factual considerations of the extrinsic and intrinsic tests)." *Id.*

300. Observing that this expert's opinion of *Whelan* had been in part responsible for *Computer Associates'* rejection of that standard, the *Gates Rubber* court nonetheless declined to "second guess Dr. Davis' understanding of [that] holding." *Id.* at 1512 n.11.

ent expert.[301] The experts found: identical constants; substantial similarities in the programs' factual findings were that the menus, formulas, data flow, control flow, install files, behavior of Engineering Calculation modules, fundamental tasks, sorting criteria (organization of data), and common errors/misbehaviors; similarities in the "look and feel," in levels of complexity, and in the overall structure and organization of Design Modules; differences in programming style; but no similarity or evidence of copying of object or source code.[302] The court agreed with Dr. Davis that Bando American's Chauffeur program had been copied from Gates Rubber's Data Flex program.[303]

As for the intrinsic test, Judge Sparr, who acknowledged "being largely unfamiliar with computers and their processes," observed at a demonstration "significant similarities in the running of the two programs: while the appearance of the screens was different, the content and method of proceeding through calculations were quite similar, as was the overall operation of the two programs."[304]

In applying the *Computer Associates* test, the court arrayed various elements of the programs along the "idea/expression" continuum, ranging from the "look and feel" of the program, which was more connected with the "idea" of the program, to the engineering module[305] and the data flow and control flow,

---

301. *Id.* at 1514. Dr. Clancey's comparison points between the two programs were "the identity of the programmer; whether the material was unpublished; the relationship between concepts and constants; the population of similar programs in the world; the derivability of portions of the programs; and the range, space and alternative designs." *Id.* Dr. Davis's analysis focused on similarities between the programs' "arrangement, the types of choices available and their presentation; data flow, which he described as being analogous to 'recipe,'; control flow, which is the sequence of events; and the sequence, which is the sequence of behaviors and internal functions." *Id.* Dr. Dorn's testimony was largely confined to the means by which the two programs could have been independently developed. *Id.* at 1515.

302. *Id.* at 1516.

303. *Id.*

304. *Id.*

305. *Id.* at 1518. This module

> falls closer to the expression range because, although the pure engineering aspect in a broad sense may be more likely to be not protected, the relevant engineering modules in the two programs contain particular elements that perform in similar manners. The [module contain-

which were closer to expression.[306] Diverging from *Computer Associates*, the court also held that copyright protection could extend to a program's behavior - in this case, its "common errors and misbehaviors."[307]

Although the formulas behind Design Flex, which appeared in identical form in Chauffeur, were not protectable because they had been previously published,[308] the court found that the appearance in Chauffeur of "numerous" protectable portions of Design Flex, and in particular of the mathematical constants, which no other program except Chauffeur had a need to duplicate, implicated Bando American in the copyright infringement of Gates Rubber's program.[309]

Thus, the court issued an injunction restraining Bando from using or distributing Chauffeur and ordering it to retrieve all existing copies of that program from the hands of third parties for return to the court and to return to Gates Rubber all copies of the Design Flex program in Bando's possession.

B. *The* Lotus *Trilogy: Reconciling* Computer Associates *with* Paperback

In *Lotus Development Corp. v. Borland International, Inc.*,[310] the latest in a series of rulings on Lotus's copyright infringement claims against Borland,[311] the District Court of Massachusetts

---

ing algorithms for belt design] is somewhat more difficult as it involves algorithms, or procedures for solving given types of mathematical problems.

Id.

The court declined to hold that the algorithms were not protectable by copyright law. *Id.*

306. *Id.*

307. *Id.* "A particular example of common error concerns the minimum/maximum error where both programs, upon receiving a particular answer, erroneously take the user back to another part of the program," thus revealing the interconnection of different parts of the program. *Id.* The court found that this error was "part of the creative expression of the program itself." *Id.* at 1518-19.

308. *Id.* at 1519.

309. *Id.*

310. 799 F. Supp. 203 (D. Mass. 1992).

311. *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (D. Mass. 1992) (Memorandum and Order holding that Lotus did not properly frame its allegations that Borland had infringed copyright on elements of user interface);

granted in part Lotus's motion for summary judgment and denied Borland's motion for summary judgment. In concluding that Borland's spreadsheet program Quattro Pro infringed on the copyright of Lotus 1-2-3 (1-2-3), the court equated its own three-part test for software infringement with the analysis of *Computer Associates*, which, like *Whelan* itself, had not been intended to cover screen displays.[312]

1. *Lotus Development Corp. v. Paperback Software International* (1990)

In *Lotus Development Corp. v. Paperback Software International*,[313] Judge Keeton extensively analyzed the idea/expression distinction before concluding that the defendant's program, VP-Planner, had infringed on 1-2-3. After reviewing the existing case law and commentary in depth, the court advanced the following "statement of the most significant elements of the legal test for copyrightability":[314]

> FIRST, in making the determination of "copyrightability," 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. . . .

Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990) (user interface of Lotus 1-2-3 is copyrightable).

312. "As a caveat, we note that our decision here does not control infringement actions regarding categorically distinct works, such as certain types of screen displays. These items represent products of computer programs, rather than the programs themselves, and fall under the copyright rubric of audiovisual works. If a computer audiovisual display is copyrighted separately as an audiovisual work, apart from the literary work that generates it (*i.e.*, the program), the display may be protectable regardless of the underlying program's copyright status." Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 703 (2d Cir. 1992). "In this case . . . we are concerned not with a program's display, but the program itself, and then with only its non-literal components." *Id.*

Cf. Manufacturers Technologies, Inc. v. CAMS, Inc., 706 F. Supp. 984 (D. Conn. 1989) (applying extrinsic/intrinsic test to find copyright infringement of screen displays).

313. 740 F. Supp. 37 (D. Mass. 1990).

314. *Id.* at 60.

> SECOND, 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.
>
> THIRD, 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."[315]

Under the second element of this test, the court compared such spreadsheet programs as 1-2-3, Visicalc, and Excel, and determined that the "rotated 'L' " configuration of the screen display was not only common but indeed one of only a few ways of portraying a spreadsheet on a computer screen.[316] Similarly, a popular and practical feature involving "the designation of a particular key that, when pressed, will invoke the menu command system" could be implemented in a limited number of ways, *i.e.*, by using one of the few keys not reserved for numbers, letters and mathematical operations. The copyrightability of choosing the slash key "/" for this operation was therefore disfavored, as was the use of the keys "+", "-", "*" and "/" (within formulas) to indicate addition, subtraction, multiplication, and division, respectively.[317]

However, the court held that a program's visual presentation of the various memo commands in a "main menu command line":

---

315. *Id.* at 60-61.

316. *Id.* at 66. However, the idea of using a "two-line moving cursor," while "functional and obvious" and common, did not automatically render a menu system noncopyrightable, since a developer might be able to add to this feature "substantial elements of expression, distinctive and original, which are thus copyrightable." *Id.* at 65.

> A two-line moving cursor "presents the user with a list of command choices (*e.g.*, 'file', 'copy', 'quit') and a moving cursor to use in communicating ('entering') the choice. . . . The top line of the menu contains a series of words, each of which represents a different command. . . . The second line of the menu displays a 'long prompt,' which contains further information about the highlighted command."

Id. at 63-64.

317. *Id.* at 66-67. The court also noted that to ease the user's access to this feature, "the user should not be required to press two keys at the same time" to invoke the command system. *Id.* at 66.

> is not essential to the electronic spreadsheet idea, nor does it merge with the somewhat less abstract idea of a menu structure for an electronic spreadsheet. The idea of a menu structure — including the overall structure, the order of commands in each menu line, the choice of letters, words or "symbolic tokens" to represent each command, the presentation of these symbolic tokens on the screen (*i.e.*, first letter only, abbreviations, full words, full words with one or more letters capitalized or underlined), the type of menu system used (*i.e.*, one-, two- or three-line moving-cursor menus, pull-down menus, or command-driven interfaces), and the long prompts — could be expressed in a great many if not literally unlimited number of ways.[318]

Thus, the menu structure was copyrightable under the second step of the court's copyrightability test. The distinctiveness of the "structure, sequence and organization of the menu command system" of 1-2-3, as well as the trouble that Paperback Software and codefendant Stephenson Software, Ltd. had taken to copy it, satisfied the third element of the *Paperback* copyrightability test as well.[319]

2. *Lotus Development Corp. v. Borland International, Inc.*, 788 F. Supp. 78 (D. Mass. 1992)[320]

Over two years later, the court addressed similar issues in the context of Lotus's allegations that Borland's spreadsheet programs Quattro and Quattro Pro infringed on 1-2-3. At issue was the copyrightability of the "user interface" of 1-2-3, which "includes 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.' "[321]

In dividing the issues between the judge and jury, Judge Kee-

---

318. *Id.* at 67.

319. *Id.* at 68.

320. 788 F. Supp. 78 (D. Mass. 1992).

321. 740 F. Supp. 37, 63 (D. Mass. 1990) (quoting Lotus's Post-Trial Brief at 53). Several of these elements, and other relevant terms, were themselves defined in Lotus Dev. Corp. v. Borland Int'l, 799 F. Supp. 203, 206 (D. Mass. 1992):

> "Command" refers to an abbreviated description of a direction that a user of a software program (whether Lotus 1-2-3, Borland's Quattro

ton questioned whether "substantial similarity" and "copyrightability" were issues "in which law and fact are so deeply intertwined that, at least as a practical matter if not strictly in principle as well, total separation cannot be achieved."[322] The court denied the cross-motions for summary judgment, instead allowing the parties to restate their motions in light of the clarifi-

---

Pro, or another program) may invoke to cause some operation to be performed.

"Menu" refers to a display on the computer monitor of a limited number of commands available to the user at a given moment.

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

"Command structure" refers to the organization of the menus and menu commands. (Other phrases used with essentially the same meaning include "menu command structure," "menu hierarchy," and "menu command hierarchy.") In Lotus 1-2-3, menu commands are organized so that less than a dozen related menu commands are displayed at any given moment. This display communicates to the user the spreadsheet operations immediately available. Each menu of less than a dozen commands is linked to the preceding/succeeding menus by the operation of menu commands. All command menus are ultimately linked to a singly main (root/trunk) menu to form a "menu tree."

"Keystroke sequence" refers to a sequence of keystroke entries that a user may invoke. Keystroke sequences may be generated as one navigates the menu command hierarchy performing sequential spreadsheet operations.

"Long prompt" refers to a displayed multi-word English-language description of a "highlighted" menu command. A "highlighted" menu command appears on the computer monitor as a block of inverse video - that is, on a monochrome monitor with a black background on which characters are lit, a highlighted word appears as black letters within a lit block.

"Macro language" refers to a feature by which a user may define a very short keystroke sequence as equivalent to a longer keystroke sequence. Thus, a user may invoke the short keystroke sequence (a "macro") as a substitute for the longer keystroke sequence.

322. 788 F. Supp. at 83-84. The *Paperback* analysis of giving copyrightability issues to the jury had been mooted by the stipulation of the parties, in agreeing to a phased trial involving no jury for Phase One, that the court alone decide such fact questions in that stage of the proceedings. *Id.* at 94-95.

cation set forth in the opinion.[323]

The court also revised the *Paperback* three-part test in two ways. First, in response to an amicus brief filed by eleven professors of intellectual property law, the court substituted references to " *'idea,' 'system,' 'process,' 'procedure,' or 'method'* " for the term "idea," to accentuate "the difference between a useful process and an original expression."[324] In addition, the third part of the test was amended to indicate that the inquiry would focus on whether non-essential expressive elements, "taken together," constituted a substantial part of the allegedly infringed work, and were thus copyrightable.[325] Thus revised, these guidelines formed one suggested jury question (of mixed fact and law) on copyrightability.[326]

Yet even this formulation left juries unaccountable for their reasoning, and "free as a practical matter to reach decisions inconsistent with the balance struck by Congress, as interpreted by the courts," thus inevitably making such verdicts unpredictable[327] and the litigations longer, more complex and more expensive.[328] Since neither the Copyright Act nor the Constitution mandates that juries decide factual issues relating to copyrightability, the court held that "at least in the circumstances of this case (and probably more generally, though I need not so determine here), the issue or issues of copyrightability, including any fact questions bearing upon them, must be determined by the court, not the jury."[329] However, the court gave the parties the opportunity to submit clarified forms of instructions on

323. Though generally judges decide issues of law and juries issues of fact, "mixed" issues of law and fact demand one of two special procedures. *Id.* at 82-83. The court could first decide the issues of law and then submit the relevant factual questions to the jury for decision embedded in mixed questions of law and fact. *Id.* The jury decision then constitutes a "special verdict" under the Federal Rules of Civil Procedure 49(a) or a "general verdict accompanied by answers to interrogatories" under Rule 49(b). *Id.* Alternatively, the court could submit only fact questions to the jury. Such questions qualify as "special questions" under Rule 49(a) or as "interrogatories" under Rule 49(b). *Id.*

324. *Id.* at 90 (emphasis in original).

325. *Id.*

326. *Id.* at 93.

327. *Id.* at 95.

328. *Id.* at 96.

329. *Id.*

copyrightability to the jury.[330]

Significantly, Judge Keeton rejected Borland's argument (which would later be made by Computer Associates) that the determination of "copyrightability," that is, an identification of the protectable elements of the plaintiff's program, should precede the inquiry into whether the defendant had copied such an element or elements. Although *Concrete Machinery* had adopted the approach recommended by *Borland*,[331] the *Lotus* court found that "pattern of analysis and decisionmaking . . . is not meant to be a straitjacket [and] need not be applied to a case in which there are compelling practical reasons for a different order or proceeding."[332] The court thus suggested a verdict form asking first whether the jury found that the Quattro Pro user interface as a whole was copied from the 1-2-3 user interface as a whole; next asking whether a specific part of the Quattro Pro user interface (the "emulation interface" or the "macro facility") as a whole was so copied; and then asking which part of the copied interface or facility had been copied into some part of Quattro Pro.[333]

3. *Lotus Development Corp. v. Borland International, Inc.*, 799 F. Supp. 203 (D. Mass. 1992)

Lotus Development Corp. v. Borland International, Inc.,[334] afforded the court several opportunities to observe that its own *Pa-*

---

330. *Id.*

331. As a preliminary matter, the court can 'dissect' the copyrighted work to identify those aspects of the expression that are not necessarily mandated by the idea it embodies. . . . By dissecting the accused work and identifying those features which are protected in the copyrighted work, the court may be able to determine as a matter of law whether or not the former has copied protected aspects of the latter. The court can also determine, in at least a general way, those aspects of the work that are protected by the copyright and that should be considered in the subsequent comparative analysis under the ordinary observer test. Assuming copying of protected aspects is established, the trier of fact can then assess pursuant to the "ordinary observer" test whether there is substantial similarity between the protected expression and the accused work.

Concrete Mach. Co. v. Classic Lawn Ornaments, Inc., 843 F.2d 600, 608-09 (1st Cir. 1988) (citations omitted).

332. *Borland*, 788 F. Supp. at 86.

333. *Id.* at 87-88.

334. 799 F. Supp. 203 (D. Mass. 1992).

perback test for "copyrightability" was "compatible substantively though different in methodology"[335] from the abstraction/filtration/comparison test of *Computer Associates*.[336] The "abstraction" process, which corresponds to the *Lotus* first step of identifying a "formulation . . . for the purpose of distinguishing between the idea and its expression," was similarly derived from Judge Hand's opinions.[337] Moreover, Judge Keeton's second step, which determined whether a work that was allegedly copyrightable as an expression contained elements not required to express that idea, conformed to the *Computer Associates* "filtration" process.[338] Finally, the third *Lotus* step, which concerned whether the unessential elements were a substantial part of the allegedly infringed work, was included in the *Computer Associates* "comparison" process. The Second Circuit had added only a further comparison of the two works to determine whether the allegedly infringing work had duplicated a substantial part of the plaintiff's work's copyrightable elements.[339]

Indeed, Judge Keeton noted that the Second Circuit had not taken significant issue with *Paperback* or with the March 20 *Lotus* Order.[340] Nor had the Second Circuit's rejection of *Whelan*

335. *Id.* at 212.

336. The court found no "substantive implications . . . and . . . especially, not for this case" in the Second Circuit's description of the *Computer Associates* test as one for "substantial similarity" rather than for "copyrightability." *Id.* at 215.

337. *Id.* at 211-12.

338. *Id.* at 212.

339. *Id.*

340. The Second Circuit had observed that "[w]hile incentive based arguments in favor of broad copyright protection are perhaps attractive from a pure policy perspective, [citing Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 58 (D. Mass. 1990)], ultimately, they have a corrosive effect on certain fundamental tenets of copyright doctrine." *Computer Assocs.*, 982 F.2d 693, 712 (2d Cir. 1992) (criticizing an incentive-based reason stated at one point in the *Paperback* opinion). "The criticized argument, however, was by no means essential to the outcome in *Paperback*, and acceptance or rejection of that outcome is not likely to affect the outcome in this case." *Borland*, 799 F. Supp. at 212.

Similarly, the district court noted that its application of the merger doctrine to extend copyright protection to "certain non-literal, noncode (nonstructural) aspects of the 1-2-3 spreadsheet in *Paperback*" had been approved by the Second Circuit. *Borland*, 799 F. Supp. at 214 (citing *Computer Assocs.*, 982 F.2d at 709).

affected the validity of *Paperback*: not only was *Computer Associates* not binding authority in the First and Third Circuits, but also, unlike *Whelan*, *Paperback* had recognized that one computer program (or, in that case, the 1-2-3 interface) could contain more than one idea.[341]

The court rejected Borland's claim that the 1-2-3 interface was designed around the nature of the user macros with which it would interact, and was thus not copyrightable under the *Computer Associates* filtration test. Although program elements dictated by the specifications of preexisting hardware or software would not be copyrightable under the Second Circuit's test, the court determined as a matter of fact that "[t]he Lotus 1-2-3 interface — or at least a version of it — was written first. All user macros derive from it."[342] In fact, although the keystrokes assigned by Lotus to its various macros were "necessarily" duplicated by later programs seeking compatibility, the first set had been chosen arbitrarily.[343]

Just as the court's March 20 Order[344] had not been contained by the order of the copyrightability test enunciated in *Concrete Machinery*, so did Judge Keeton's July 31 decision decline to view either the First Circuit's approach or "the Second Circuit's three-step test . . . as a rigid barrier to alternate methods of analysis and decision."[345] The court therefore initiated the copyrightability analysis despite unresolved factual issues concerning elements of the interface and the degree to which the interface design had been dictated by functional considerations.[346]

---

341. *Borland*, 799 F. Supp. at 215. *See Paperback*, 740 F. Supp. at 64-65 (analyzing copyrightability of idea of electronic spreadsheet, of two-line moving cursor, and of specific macro commands); *Borland*, 799 F. Supp. at 215 ("[T]he Second Circuit criticized the *Whelan* decision for reasons that in large part do not apply to the rulings I made in *Paperback* or to the rulings I have made and now make in this case.").

342. 799 F. Supp. at 213. Nor did the court accept Borland's argument that the assignment of different individual letters to the various commands on each menu was a "functional constraint"; "other symbolic tokens," such as numbers, could be used. *Id.*

343. *Id.* at 213-14.

344. Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (D. Mass. 1992).

345. 799 F. Supp. at 215.

346. *Id.* at 216.

In applying the first step of its test, the court set forth five alternative formulations of 1-2-3.[347] Immediately, it rejected the most specific conception of the program's "idea," which correlated each menu command to others: the adoption of this formulation would deny copyrightability to the entire interface, leaving the way clear for Borland legitimately to incorporate the entire menu structure of 1-2-3 into its own products.[348] By also discarding the most abstract formulation of 1-2-3, as "an electronic spreadsheet," because it set "an inappropriately abstract boundary between idea and expression," the court championed the analysis of *Computer Associates* over that of *Whelan*.

Determining that "the selection of functional operations that the spreadsheet performs must be considered part of the idea of the program,"[349] the court selected the second most specific op-

---

347. In order, from the most abstract to the most particular, the selections were:

(1) Lotus 1-2-3 is an electronic spreadsheet.

(2) It is a menu-driven electronic spreadsheet.

(3) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command.

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

(5) Finally, one may conceive of the interface as that precise set of menu commands selected by Lotus, arranged hierarchically precisely as they appear in 1-2-3. Under this conception, the interface comprises the menu of commands "Worksheet," "Range," "Copy," "Move," "File," "Print," "Graph," "Data," "System," and "Quit," linked by operation of the command "Worksheet" to the menu of commands "Global," "Insert," "Delete," "Column," "Erase," "Titles," "Windows," "Status," and "Page," etc. (The completion of this proposed statement of the "idea," listing all of the more than 400 commands for which "etc." stands, would require several dozen more lines of text.)

Id.

348. *Id.* at 217.

349. *Id.*

tion for a formulation of the 1-2-3 user interface, which described the structure of the hierarchical system of menus which among them granted access to all of the 1-2-3 spreadsheet operations.[350]

In the second step of the process, the court identified the specific menu commands and command structure of 1-2-3 as elements of expression. Not only had Borland actually created a variant of this structure for incorporation into Quattro Pro, but Lotus' menu and submenu commands themselves (*e.g.*, "Worksheet," "Range," "Copy," "Move," "File," and "Print") could easily have been renamed to create "a sufficient alternate method of implementing the system."[351] Finally, the apparent ordering of the menu commands "in order of the expected frequency of use" was not so dictated by functional concerns that "at least hundreds and perhaps thousands" of alternative expressions of the commands were not still possible.[352]

Under the third step, the court found that the menu commands, menu command hierarchy, macro languages and keystroke sequences were a substantial part of 1-2-3, and indeed that "[n]o reasonable jury could find otherwise."[353] "That is, a reasonable fact-finder must conclude that the Quattro programs derive from illicit copying. The emulation interfaces are substantially similar in the mixed law-fact sense to the Lotus 1-2-3 user interface."[354]

The court scheduled a jury trial on the issue of whether the long prompts of the 1-2-3 interface had been infringed by Quattro

---

350. *Id.* *See supra* note 347 (Option (4)).

351. *Id.* at 218.

352. *Id.* The court also observed: (a) that the ordering would have reflected, at best, only Lotus's *predictions* of the frequency of the command use; (b) that the validity of such predictions would depend on the user and his or her uses of the spreadsheet; (c) an apparent exception in this ordering; and (d) the vagueness of the "order" of commands in "wraparound" horizontal menus through which users could move either left or right. *Id.* at 218-19. Thus, "the arrangement of menu commands according to predicted frequency of use is not a major functional limitation on the number of arrangements of menu commands." *Id.* at 219.

By contrast, a jury might find the grouping of menu commands by function a functional concern, although the resolution of this factual issue would only affect the scope of Lotus's relief. *Id.* at 218.

353. *Id.* at 219.

354. *Id.* at 221.

Pro. Since the court had already held, after comparing the two programs as a whole, that Quattro Pro infringed 1-2-3, the only questions left unresolved, in the determination of damages, were (a) whether (in the view of the jury) Borland copied the long prompts as well and (b) whether (in the eyes of the court) the long prompts were copyrightable expressive elements of 1-2-3. "There would be no need to ask separately whether the copying of the long prompts would alone, or in combination, render the Quattro programs substantially similar to Lotus 1-2-3."[355]

C. *Unlocking the Door: Reverse Engineering*[356]

Though not condoning the incorporation into Altai's products of Computer Associates' source code for ADAPTER, the Second Circuit appeared to legitimize Altai's "clean room" stratagem for removing the pirated code from OSCAR 3.4.[357] Two recent circuit court decisions in the context of video games have approved competitors' use of technological methods to extract the code of "lockout" or "lock and key" subprograms from commercially available products of their rivals[358] in order to achieve compati-

---

355. *Id.*

356. "Reverse engineering" has been defined by the Supreme Court as "a fair and honest means [of] starting with the known product and working backward to divine the process which aided in its development or manufacture." Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 476 (1974).

In the software context,

> *[d]isassembly* is the process of translating a machine language program into an assembly language program; *decompilation* is the process of translating a machine language program into a high-level program. . . . There are currently no commercially available decompilers. It appears that the term "decompilation," as it is used in the policy debate, encompasses disassembly and any other procedure by which a machine language program is translated into a more understandable form. There are a number of disassembler programs available on the market.

FINDING A BALANCE, *supra* note 3, at 7.

357. *See supra* notes 72, 177.

358. The OTA's report suggested that, to resolve issues concerning reverse engineering, Congress could: modify the scope and subject matter of copyright protection for software; direct the Copyright Office to develop guidelines for the "fair use" of software and the "essential steps in the utilization" of programs; or legislatively specify the status of reverse engineering under the "fair use" doctrine. FINDING A BALANCE, *supra* note 3, at 31.

bility.[359] Not only do these endorsements of "reverse engineering" rely on the analysis of *Computer Associates*, but developers obtaining source or object codes in this manner will clearly be encouraged to subject them to the abstraction/filtration/comparison test to gauge the limits of permissible similarity. Notably, among the elements of the original program that would be "filtered out" as not protectable under *Computer Associates* are those determined by such external factors as "compatibility requirements of other programs with which a program is designed to operate in conjunction" and "the demands of the industry being serviced."[360]

In holding that Atari had infringed on the "lock and key"[361] program contained in Nintendo's home video game system, the Court of Appeals for the Federal Circuit noted *Computer Associates*'s and *Brown Bag*'s agreement on "separating the program [at issue] into manageable components [to ease] the court's task of discerning the boundaries of protectable expression"; it then applied the Second Circuit's filtration analysis to these components.[362]

---

359. *See, e.g.*, Gary R. Ignatin, *Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility*, 140 U. PA. L. REV. 1999, 2023 (1992) (one case in which reverse engineering is particularly justified is in making new program compatible with existing copyrighted software).

360. Computer Assocs. Int'l., Inc. v. Altai, Inc., 982 F.2d 693, 710 (2d Cir. 1992).

361. Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d. 832 (Fed. Cir. 1992).

To protect the integrity of its Nintendo Entertainment System ("NES"), which consists of a monitor, console and controls, Nintendo designed the 10NES program, which would

> prevent the NES from accepting unauthorized game cartridges. Both the NES console and authorized game cartridges contain microprocessors or chips programmed with the 10NES. The console contains a "master chip" or "lock." Authorized game cartridges contain a "slave chip" or "key." When a user inserts an authorized cartridge into a console, the slave chip in effect unlocks the console; the console detects a coded message and accepts the game cartridge. When a user inserts an unauthorized cartridge, the console detects no unlocking messages and refuses to operate the cartridge.
>
> Nintendo's 10NES program thus controls access to the NES.

Id. at 836.

362. *Id.* at 839.

The basic principle of Nintendo's program, that is, the generation of a stream of data as a "key" to "unlock" the console, was found to be an unprotectable idea.[363] "After filtering [such] unprotectable elements out of the 10NES program,"[364] the court held that the specific "key" implemented by Nintendo was copyrightable as an expression, since many different such streams could have been used.[365]

Significantly, because the object code on the Nintendo computer chip was not accessible or comprehensible without reverse engineering,[366] the court upheld Atari's "peeling" of the Nintendo 10NES chips[367] as a "fair use" under the Copyright Act.[368] However, "[t]his fair use did not give Atari more than the right to understand the 10NES program and to distinguish the protected from the unprotected elements of the program. Any copying beyond that necessary to understand the 10NES

363. *Id.* at 840.
364. *Id.*
365. *Id.*
366. *Id.* at 843.
367. Reverse engineering, untainted by the purloined copy of the 10NES program and necessary to understand 10NES, is a fair use. An individual cannot even observe, let alone understand, the object code on Nintendo's chip without reverse engineering. Atari retrieved this object code from NES security chips in its efforts to reverse engineer the 10NES program. Atari chemically removed layers from Nintendo's chips to reveal the 10NES object code. Through microscopic examination of the "peeled" chip, Atari engineers transcribed the 10NES object code into a handwritten list of ones and zeros. While these ones and zeros represent the configuration of machine readable software, the ones and zeros convey little, if any, information to the normal unaided observer. Atari then keyed this handwritten copy into a computer. The computer then "disassembled" [*i.e.*, reconstructed into "source code"] the object code or otherwise aided the observer in understanding the program's method or functioning.

Id. at 843-44.

368. " '[F]air use of a copyrighted work, including such use by reproduction in copies . . . for purposes such as criticism, comment, news reporting, teaching . . . scholarship or research' is not infringement." 17 U.S.C. § 107, *quoted in*

program was infringement."[369] The similarities between the programs, excluding the idea of the "key" data stream, supported infringement; for example, Atari's chip contained not only features of the 10NES program that were irrelevant to its own program, but unnecessary instructions that Nintendo had subsequently deleted from the program.[370]

Adopting the *Computer Associates* analysis "in light of the essentially utilitarian nature of computer programs," *Sega Enterprises Ltd. v. Accolade, Inc.*,[371] like *Atari*, extended the fair use doctrine to permit "persons who are neither copyright holders nor licensees to disassemble a copyrighted computer program in order to gain an understanding of the unprotected functional elements of the program . . . when the person seeking the understanding has a legitimate reason for doing so and when no other means of access to the unprotected elements exist."[372] The method employed to capture the object code of the plaintiff's

---

Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 842-43 (Fed. Cir. 1992).

Section 107 identifies the following factors as relevant to a determination of "fair use":

(1) the purpose and character of such use, including whether such use is of a commercial nature or is for nonprofit educational purposes;

(2) the nature of the copyrighted work;

(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market for or value of the copyrighted work.

To invoke this exception to the standards of copyright infringement, the reverse engineering process must have been applied to an authorized, *i.e.*, legitimately obtained, copy of the plaintiff's work. 975 F.2d at 843.

369. *Id.* at 844.

370. *Id.* at 844-45.

371. 977 F.2d 1510 (9th Cir. 1992). This opinion was originally released by a three-judge panel on October 20, 1992. Following Sega's petition for rehearing and suggestion for rehearing en banc, the same panel amended the opinion on January 6, 1993 by including a footnote specifically rejecting Sega's contention that, under the logic of *Atari*, Sega's security code constituted protectable expression. *Id.* at 1524 n.7.

372. *Id.* at 1514. Because object code is generally incomprehensible to humans, disassembly was necessary to discover the compatibility requirements of the Genesis system. Even the "chip-peeling" or "clean room" techniques of reverse engineering, the court noted, involved disassembly. *Id.* at 1525-26.

video console locking mechanism differed from that in *Atari*,[373] but the result was the same. In determining that "Accolade copied Sega's code for a legitimate, essentially non-exploitative purpose," the court stressed that Atari was not seeking to shirk the effort of developing both its own programs, with their own code, for the Sega system, or to avoid the payment for use of the interface procedures for the Genesis console.[374] Indeed, Accolade's penetration of the Sega lock's defenses "has led to an increase in the number of independently designed video game programs offered for use with the Genesis console. It is precisely this growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote."[375]

Sega v. Accolade was specifically amended to reject "Sega's be-

---

373. As part of the reverse engineering process, Accolade transformed the machine-readable object code contained in commercially available copies of Sega's game cartridges into human-readable source code using a process called 'disassembly' or 'decompilation.' Accolade purchased a [Sega] Genesis console and three Sega game cartridges, wired a decompiler into the console circuitry, and generated printouts of the resulting source code. Accolade engineers studied and annotated the printouts in order to identify areas of commonality among the three game programs. They then loaded the disassembled code back into a computer, and experimented to discover the interface specifications for the Genesis console by modifying the programs and studying the results. At the end of the reverse engineering process, Accolade created a development manual that incorporated the information it had discovered about the requirements for a Genesis-compatible game. According to the Accolade employees who created the manual, the manual contained only functional descriptions of the interface requirements and did not include any of Sega's code.

[In creating its own games for the Genesis,] Accolade did not copy Sega's programs, but relied only on the information concerning interface specifications for the Genesis that was contained in the development manual. Accolade maintains that with the exception of the interface specifications, none of the code in its own games is derived in any way from its examination of Sega's code.

Id. at 1514-15.

374. *Id.* at 1523.

375. *Id.* The court also observed that the video market was not so limited that the sale of an Accolade program would significantly affect the market for the sale of a Sega program for a different type of game, or even for the same type of game (*e.g.*, Accolade's "Mike Ditka Power Football" versus Sega's "Joe Montana Football."). *Id.* at 1523-24.

lated suggestion" that the Sega lockout code, like that of Nintendo's 10NES program, was protectable as an expression.[376] Declaring its opinion "entirely consistent" with the Federal Circuit's *Atari* decision, the Ninth Circuit distinguished the 10NES program as creative and original, and as copyrightable expression of an idea — the generation of random streams of data to unlock the software system — that could be accomplished in many different ways.[377]

By contrast, the court found "Sega's key appears to be functional [and thus noncopyrightable under *Computer Associates*]. It consists merely of 20 bytes of initialization code plus the letters S-E-G-A. There is no showing that there is a multitude of different ways to unlock the Genesis III console."[378] However, this conclusion appears disarmingly facile. Even though not so elaborate as that of Nintendo, Sega's protection system expresses the idea of unlocking the Sega console by a *short* stream of data. Moreover, S-E-G-A is hardly the only four-letter combination that could have been selected as part of this "key."

As with *Whelan*, and with *SAS* and *Q-Co*, equity considerations underlie the distinction between *Atari*, which upheld Nintendo's preliminary injunctions against Atari on grounds of copyright infringement, and *Sega*, which lifted Sega's similar injunction against Accolade.[379] To obtain a reproduction of the 10NES program from the Copyright Office, "Atari [had] falsely alleged that it was a present defendant in a case in the Northern District of California [and represented to] the 'Library of Con-

376. *Id.* at 1524 n.7.

377. *Id.* (citing Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 839 (Fed. Cir. 1992)).

378. *Id.* The court also suggested that the small size of the Sega security code deprived the code of protection, under the "words and short phrases doctrine." *Id.* (citing 37 C.F.R. § 202.1(a)).

379. Richard Raysman & Peter Brown, *Recent Software Copyright Decisions*, N.Y. L.J., Oct. 13, 1992 at 3, 5. However, the original *Sega* opinion's application of the "fair use" doctrine has been attacked on the grounds, *inter alia*, that: (a) Accolade directly damaged Sega by reverse-engineering rather than paying license fees for the Sega copyrights; (b) the exact lockout code used by Sega, as one among many possible combinations of 1's and O's, was a protectable expression; and (c) other methods of reverse engineering do not create unauthorized interim copies. Martin Glenn & Dale Cendali, *Sega Case Suggests Protection Strategies*, NAT'L. L.J., Jan. 18, 1993, at S2, S6.

gress that the requested copy [would] be used only in connection with the specified litigation.' "[380] By contrast, where Accolade had made no such misrepresentation and stood ready to invest its own original efforts in producing programs for a competitor's game system, Sega did not satisfy the court that the trademark security system in the Sega products, which also invoked a screen display of Sega's trademark and a message that the game was produced by Sega, was "nonfunctional" and thus protectable.[381]

VI. CONCLUSION

Reconnoitering "the vast landscape of possible 'tests' for substantial similarity"[382] of computer software should be much simplified by the Second Circuit's landmark "abstraction/ filtration/ comparison" analysis. *Computer Associates v. Altai* combines copyright's idea/expression distinction, *scenes a faire* and merger doctrines, and substantial similarity tests into an approach suitable for assessing works that, unlike books and plays, have literary aspects but are primarily functional.

Noteworthy in the Second Circuit's rejection of *Whelan* is the court's detailed examination of a program's various levels of abstraction, from the "literal" levels of object code and source code to the non-literal "structure, sequence and organization" of the subprogram's arrangement and interaction. Elements of a program will not merit copyright protection if they are dictated by the nature of the program or by the necessary hardware, or if they are in the public domain. Moreover, efficiency considerations will remove other valuable features of software from the "golden nugget" of a program's copyrightable material.

By their extensive analyses of the plaintiff's product, the *Computer Associates* and *Lotus v. Borland* tests identify, and indirectly invite the duplication of, the noncopyrightable elements of market-leading programs. The dissemination of these features

380. *Atari*, 975 F.2d at 836 (quoting Atari's application to Copyright Office for reproduction of 1ONES program).

381. Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510, 1530-32 (9th Cir. 1992).

382. Gates Rubber Co. v. Bando Am., Inc., 798 F. Supp. 1499, 1511 (D. Colo. 1992).

may ultimately benefit not only the consumer but also the software industry itself. This trend can only be encouraged by the increased judicial tolerance of such reverse-engineering techniques as "clean room" programming, chip "peeling" and decompilation.

Nonetheless, various risks arise from the evidentiary aspects of the Second Circuit's approach. Although the complex issues of software design, development and programming demand expert testimony, such evidence may be subtly biased or may range into the area of legal analysis, even when introduced by a court-appointed expert.

Once Congress or the Supreme Court clarifies the issues addressed by *Computer Associates*, patent law may emerge as the preferred mode of protection of computer programs. Until that day, however, the appropriateness and effectiveness of copyright protection for software's non-literal elements will undoubtedly remain in flux.