Notre Dame Law School

NDLScholarship

Journal Articles

Publications

1994

The Copyrightability of Nonliteral Elements of Computer Programs

Julian Velasco

Notre Dame Law School, jvelasco@nd.edu

Follow this and additional works at: https://scholarship.law.nd.edu/law_faculty_scholarship

Part of the [Computer Law Commons](#)

Recommended Citation

Julian Velasco, *The Copyrightability of Nonliteral Elements of Computer Programs*, 94 Colum. L. Rev. 242 (1994).

Available at: https://scholarship.law.nd.edu/law_faculty_scholarship/392

This Article is brought to you for free and open access by the Publications at NDLScholarship. It has been accepted for inclusion in Journal Articles by an authorized administrator of NDLScholarship. For more information, please contact lawdr@nd.edu.

THE COPYRIGHTABILITY OF NONLITERAL ELEMENTS OF COMPUTER PROGRAMS

INTRODUCTION

The goal of copyright law is "[t]o promote the Progress of Science and Useful Arts."¹ It is premised on the assumption that "encourag[ing] . . . individual effort by personal gain is the best way to advance public welfare through the talents of authors"² In order to "promote progress," however, copyright law must respond to changes in technology. Although historically Congress has done a commendable job in amending the copyright law to keep pace with technological advances, this has not been true in the area of computer technology, perhaps the most significant scientific advancement of modern times. As a result, the federal courts have assumed the role of keeping the copyright law up-to-date in this area.³ Although the courts were able to resolve the early, simpler issues quickly and easily, they have had difficulty resolving the more complex issues that have arisen as the computer industry has developed.

One issue that has been problematic for the courts is whether and to what extent the nonliteral elements of computer programs are copyrightable. Nonliteral elements are aspects of the computer program other than the written code itself. For example, the structure of the program, like the plot of a story, is a nonliteral element; the user interface⁴ of the program is another. Although it is clear that copyright protection extends to the literal elements of a computer program (the code), it is unclear to what extent protection extends to nonliteral elements. Because entirely different lines of computer code could employ the same program structure or create very similar user interfaces, programmers have a clear interest in having nonliteral elements of their programs protected. Judges, however, have had difficulty in determining the legitimacy of this interest under the copyright laws.

The federal courts have articulated four distinct tests for determining whether nonliteral elements of a computer program can be protected by a copyright in the program. All four tests are based on the idea-

1. U.S. Const. art. I, § 8, cl. 8.

2. *Mazer v. Stein*, 347 U.S. 201, 219 (1954).

3. The courts have interpreted Congress' inaction to have effectively shifted to them the duty of keeping the copyright law current: "Rather than itself drawing the boundary line between copyrightable and non-copyrightable elements of computer programs, Congress has mandated that courts use an evaluative standard in determining this boundary line" *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 53 (D. Mass. 1990).

4. For a definition of the term "user interface," see *infra* notes 29-32 and accompanying text.

expression dichotomy that is the foundation for copyright law,⁵ yet they are all significantly different from each other. In 1986, in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*,⁶ the Third Circuit became the first Circuit Court of Appeals to address this issue. In 1990, the Federal District Court of Massachusetts addressed the same issue but developed its own approach in *Lotus Development Corp. v. Paperback Software International*.⁷ In 1992, in the case of *Brown Bag Software v. Symantec Corp.*,⁸ the Ninth Circuit applied its general test for substantial similarity, the Extrinsic-Intrinsic Test,⁹ in the computer program context. Finally, in 1992, the Second Circuit developed its own test in *Computer Associates International, Inc. v. Altai, Inc.*¹⁰ The debate over these tests has centered on whether a given approach "promotes progress" by providing an adequate level of copyright protection, or whether it stifles creativity by providing either too much or too little protection.

This Note argues that the *Altai* test, in modified form, is the best test for determining substantial similarity—and hence copyrightability—because it most closely adheres to general principles of copyright law. Such principles have served copyright law well in other areas and were intended by Congress to be applied to computer programs as well.¹¹ Part I explains aspects of computer technology pertinent to this issue and provides a brief history of copyright law in the computer program context. Part II examines the *Whelan*, *Paperback*, *Brown Bag*, and *Altai* cases and provides a review of the criticism that has been leveled against each. Part III analyzes these tests, concludes that a modified version of the most sophisticated approach—the *Altai* test—would be best, and proposes an additional step for the analysis under that test. Finally, it reexamines

5. The idea-expression dichotomy is simply that copyright law extends protection only to the particular expression of an idea, but never to the idea itself. See *infra* text accompanying notes 69–70.

6. 797 F.2d 1222 (3d Cir. 1986).

7. 740 F. Supp. 37 (D. Mass. 1990). This district court case is significant because of the stature of the plaintiff, the developer of the very popular 1-2-3 spreadsheet program, and because of the holding, which extended copyright protection to the user interface of that program, see *infra* section II.B.

8. 960 F.2d 1465 (9th Cir. 1992).

9. The test was already developed in existing case law. It was originally derived from the Second Circuit's test announced in *Arnstein v. Porter*, 154 F.2d 464, 468–69 (2d Cir. 1946) (popular song). It was first adopted by the Ninth Circuit in the case of *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1162–65 (9th Cir. 1977) (children's television characters). The test was significantly modified in *Shaw v. Lindheim*, 919 F.2d 1353, 1356–57 (9th Cir. 1990) (television script). For a discussion of the development of the Extrinsic-Intrinsic Test, see *infra* note 178.

The Extrinsic-Intrinsic Test was also applied in one of the most significant computer program cases to date, *Apple Computer, Inc. v. Microsoft Corp.*, No. C-88-20149-VRW (N.D. Cal.): 821 F. Supp. 616 (1993); 1993 WL 207982 (1993); 799 F. Supp. 1006 (1992); 1992 WL 75423 (1992); 779 F. Supp. 133 (1991); 759 F. Supp. 1444 (1991); 717 F. Supp. 1428 (1989); 709 F. Supp. 925 (1989).

10. 982 F.2d 693 (2d Cir. 1992).

11. See *infra* text accompanying notes 62–64.

some of the major concerns that are often raised against providing any protection to nonliteral elements of computer programs.

I. COPYRIGHT PROTECTION OF COMPUTER PROGRAMS

In order to understand the issues raised by the copyrightability of a computer program's nonliteral elements, it is necessary to have a basic understanding of both computer programming and the general principles underlying copyright law. This Part provides background information intended to give the reader a working knowledge of the subject matter. First, it briefly discusses how computers work and how computer programs are developed. It then reviews the legal landscape that currently surrounds the copyrightability of computer programs.

A. *Some Technical Background on Computer Programs*

A computer is basically a very sophisticated electronic machine used to manipulate data. Since computers cannot think for themselves, however, they must be told exactly what to do. A computer program¹² is the part of a computer system that tells the computer what to do: it is basically a list of instructions that the computer executes to achieve the result desired by the programmer.¹³ Computers, however, only comprehend "machine language," a binary code consisting entirely of strings of 1s and 0s (or ONs and OFFs),¹⁴ which is awkward and difficult for people to understand. Therefore, a programmer usually writes what is known as the "source code" of a program using a "higher level" language¹⁵—e.g., BASIC, Pascal, or C. Since higher level languages are composed of words and mathematical symbols, they are more easily understood by people and are therefore easier to use. After the source code is written by the programmer, it is then translated into machine language, called "object code,"¹⁶ so that it can be understood by the computer.

12. Computer programs are also known as "software," as opposed to the physical components of the computer system (such as the monitor, the storage devices, and the computer itself), which are known as "hardware."

13. Copyright law defines a "computer program" 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 U.S.C. § 101 (1988).

14. See 5 McGraw-Hill Encyclopedia of Science and Technology, Digital Computer 262, 262 (7th ed. 1992) [hereinafter E.S.T.]. Each *binary digit*—i.e., each 1 or 0 (or each ON or OFF)—is called a *bit* of information; a string of eight bits is called a *byte*; and a string of one or more bytes is called a *word*. See Computer Systems Architecture, in 4 E.S.T., supra, at 276, 277. The computer manipulates such bits, bytes, and words to perform whatever functions the program commands. See Digital Computer Programming, in 5 E.S.T., supra, at 269, 270.

15. See Digital Computer Programming, in 5 E.S.T., supra note 14, at 269, 271 ("Programming . . . is far more awkward in binary and . . . becomes increasingly difficult as programs get longer. For these and other reasons, programmers prefer to work in languages that are at a higher level than . . . machine language . . .")

16. The process of translating source code into object code is called "compiling." See Supercomputer, in 17 E.S.T., supra note 14, at 629, 632 (defining a "compiler" as "an

There are two basic types of computer programs: "operating systems" and "applications." An operating system coordinates the interactions between the hardware and the software in a computer system.¹⁷ Applications, such as spreadsheets, word processors, or even video games, perform the functions that the user desires.¹⁸ Although much of what can be said about one should also apply to the other, this Note focuses on application programs, since they are easier to understand and are far more numerous than operating systems.¹⁹

Computers are capable of only a limited number of arithmetical and logical manipulations of data. Since they are very fast and thoroughly accurate, they are able to perform very complex operations rather quickly—but only if they are told exactly how to do so. It is the programmer who indirectly, by means of the program, walks the computer through the process of solving a problem. Developing a program demands more than merely sitting at the computer and writing the source code. A multi-step process takes the program from the original idea to the final product.²⁰ At each step, the programmer progresses along the scale of specificity, moving from the general idea to the precise instructions to be given to the computer.

First, the programmer decides what purpose the software will serve for the user.²¹ For example, the required program may be an electronic spreadsheet that performs all sorts of complex calculations on data, or it may be a word processor that assists the user in drafting documents.

Second, the programmer determines the functions that will be needed to meet the user's demands.²² For example, the user might want a spreadsheet program to perform instantly a host of mathematical, statis-

automatic translator of programs written in a high-level language . . . into the []computer's . . . machine operation codes"); see also 3 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 13.03[F], at 13-102.9 n.271 (1993) [hereinafter Nimmer on Copyright].

17. See generally Operating System, *in* 12 E.S.T., *supra* note 14, at 382, 382; see also 1 Michael D. Scott, Scott on Computer Law, § 3.11[C], at 3-47 & n.189 (2d ed. 1992 & Supp. 1993).

18. See James V. Vergari & Virginia V. Shue, Fundamentals of Computer—High Technology Law 22 (1991); see also 1 Scott, *supra* note 17, § 3.11[C], at 3-48 & n.190.

19. Applications are said to "work under" or be "compatible with" a particular operating system. Each computer will only have a limited number of operating systems. For example, the IBM PC has three major operating systems: DOS, Windows, and OS/2. However, each operating system has a great number of applications which are compatible with it.

20. Sources differ as to the exact number and content of the steps involved in developing a program. This account is drawn for the most part from Software Engineering, *in* 16 E.S.T., *supra* note 14, at 531, 531; a slightly different account is given in Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 697-98 (2d Cir. 1992); many others abound. Basically, however, the various accounts amount to the same procedure.

21. This step is called "requirements definition": "the basic question is what the user expects the software to do." Software Engineering, *in* 16 E.S.T., *supra* note 14, at 531, 531.

22. This step is called "functional specification": here, "the user requirements are converted into a specification of the functions to be performed by the proposed system." *Id.*

tical, financial, and logical functions as the data are entered into the computer. The user might also want the computer to present the results in a variety of ways. Of course, the user would also need the program to be able to save and retrieve information.²³ Thus, the program would have to be able to perform all of these functions.

Third, the programmer must "design" the overall structure of the program. This consists of breaking down the various functions of the program into subtasks that are easier for the programmer to understand. The programmer designates each function as a "subroutine," or a mini-program within the main program, and breaks down each subroutine into further subroutines as necessary until each subroutine performs a relatively simple task—i.e., one that is easily programmable. Then the various subroutines must be organized in a logical and efficient pattern. Typically, this is done in the form of a flow-chart that becomes the foundation for writing the program code.²⁴

Fourth, the programmer must "implement" the design of the program by actually writing the program's "code"²⁵ To do this, the programmer must decide upon the proper problem-solving methods, or algorithms, to use for each subroutine. The programmer then encodes the design of the previous stage and each algorithm in the programming language. The product is then compiled into object code.

Fifth, the program must be "debugged":²⁶ errors in the program must be found and corrected before the program is put to use. Since not all errors are likely to be found before the program is put to use, a final step of "maintenance"²⁷ is required to correct any hidden errors and improve any imperfections later discovered in the program. This final stage is a continuous one, since errors may be encountered at any time during the useful life of the program.

23. This is a very general account of the functions that are to be included. Before moving on to the next stage, the programmer would have to determine exactly which mathematical (e.g., addition, multiplication, root derivations, etc.), statistical (e.g., averages, regression analysis, etc.), financial (e.g., present value, amortization, etc.), and logical (e.g., comparison, sorting, etc.) functions are to be performed by the program, as well as exactly how the program will present its results (e.g., printing format, types of graphs, etc.), and every other specific function the program will perform.

24. See *Software Engineering*, in 16 E.S.T., *supra* note 14, at 531, 532.

It is also important during this stage to "determin[e] the manner in which these [subroutines] relate to one another," *id.*—for example, the "parameter list" for each subroutine must be specified. For a definition of "parameter list," see *Altai*, 982 F.2d at 697–98 ("[T]he form in which information is passed between [subroutines] . . .").

25. See *Software Engineering*, in 16 E.S.T., *supra* note 14, at 531, 532.

26. The McGraw-Hill Encyclopedia of Science and Technology does not label any stage "debugging." Instead, under its approach, errors in programming are to be corrected during the implementation stage; the fifth stage is "testing" to make sure that the program outputs are correct. Thus, the "debugging" stage—in which any errors that can be found are corrected—encompasses both the fourth and fifth steps in the E.S.T. approach. See *id.* This difference is not important as a practical matter.

27. See *id.*

As should now be apparent, writing the actual code is only a small part of developing a program; much more effort and creativity goes into conceptualizing and designing the program.²⁸ This is the essence of the "act of programming"; done well, it distinguishes a fine program (or programmer) from an adequate one. Thus, the programmer has a strong interest in copyright protection for more than merely the literal code itself.

In addition to the structure of a computer program, nonliteral elements also include what is called the "user interface" of a program. The user interface is the mechanism through which the user and the computer communicate with each other.²⁹ It includes the means for selecting the functions of the program as well as the screen displays and other forms of output that are generated for such purpose.³⁰ Most often, the user interface allows for the selection of functions by organizing them in a hierarchy of menus: the user is presented with a menu—i.e., a list—from which to make a selection as to the type of function desired; this selection typically leads to a submenu (or series of submenus in which each level describes the possible choices with greater specificity, until the final submenu is reached) that allows for the selection of the desired function.

Developing a good user interface is often considered the most difficult and important part of programming.³¹ This is because the user interface determines how easy the program is to use, which is in turn a major factor in determining whether people will want to use this program

28. See *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 56 (D. Mass. 1990) ("[T]he bulk of the creative work is in the conceptualization of a computer program and its user interface, rather than in its encoding . . ."); see also *Whelan Assocs. Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1231 (3d Cir. 1986) ("[T]he larger portion of the expense and difficulty in creating computer programs is attributable to the development of the structure and logic of the program, and to debugging, documentation and maintenance, rather than to the coding.").

29. The user interface is often referred to as the "look and feel" of the program. In *Paperback*, however, the court rejected the concept of "look and feel" as "[not] significantly helpful" in determining copyrightability, and as "conclus[ory]" in that regard. See 740 F. Supp. at 62–63.

30. See, e.g., *id.* at 63 (accepting plaintiff's definition of "user interface" as including "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" (brackets in original) (quoting Plaintiff's Post-Trial Brief at 53 (Docket No. 319))); see also Raymond T. Nimmer, *The Law of Computer Technology: Rights, Licenses, and Liabilities* ¶ 1.12, at 1–72 (2d ed. 1992) ("The most common features [of a user interface] are option menus, labels, key strokes to implement choices or commands, and text describing what the choices accomplish.").

31. "[T]he 'look and feel' of a computer software product often involves much more creativity and often is of greater commercial value than the program code which implements the product . . ." *Whelan*, 797 F.2d at 1231 (quoting *InfoWorld*, Nov. 11, 1985, at 13); see also *Paperback*, 740 F. Supp. at 56 ("[C]reating a suitable user interface is a more difficult intellectual task, requiring greater creativity, originality, and insight, than converting the user interface design into instructions to the machine.").

or a competing one.³² Thus, the holder of a copyright in a computer program is often more interested in protecting the user interface from infringement than in protecting the actual source code or object code.³³

Although programmers may have a strong interest in protecting the user interface under copyright law, the user interface is a nonliteral element of the program: entirely different programs (lines of code) could create very similar, indeed identical, user interfaces.³⁴ Thus, it is not immediately clear whether the user interface deserves copyright protection.

B. *The Copyrightability of Computer Programs*

This section provides a brief review of the legal landscape in which the copyrightability of computer programs rests. First, it gives a brief review of the statutory history and some of the well-established general principles of copyright law. Next, it turns to the legislative history on the specific subject of the copyrightability of computer programs. Finally, it reviews the early case law holding computer programs copyrightable.

1. *The Statutory and Legislative History of Copyright Law.* — Copyright law is almost exclusively federal law. The Constitution grants Congress the power "[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."³⁵ This broad delegation of authority entitles Congress to determine, within constitutional limits,³⁶ the level of copyright protection necessary to achieve the purpose of the Copyright Clause.³⁷ Prior to 1909, Congress attempted to list in the copy-

32. Cf. Raymond T. Nimmer, *supra* note 30, ¶ 1.08, at 1-61 ("In some programs, the interface is unique and important and defines the program in its market by making it more (or less) appealing to purchasers than competing programs.").

Of course, another major factor is the sheer power of the program. However, if the program is too difficult for the user, it is not very useful despite its power. See Gregory J. Ramos, Note, *Lotus v. Paperback*: Confusing the Idea-Expression Distinction and its Application to Computer Software, 63 U. Colo. L. Rev. 267, 277 (1992) ("A program that is not easily used is a program that will not be used.").

33. Cf. Raymond T. Nimmer, *supra* note 30, ¶ 1.12, at 1-72 ("[M]any programs perform identical functions, and only the user interfaces distinguish a popular program from an unpopular one. This fact provides incentives for developers of popular programs to protect their interface features against copying.").

34. See *id.* ("The user interface of a program can be copied without copying that program's code.").

35. U.S. Const. art. I, § 8, cl. 8.

36. The text of the Constitution requires that the duration of the copyright be limited. See *id.* In addition, originality is a constitutional requirement for copyrightability. See *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 111 S. Ct. 1282, 1288 (1991).

37. See *DeepSouth Packing Co. v. Laitram Corp.*, 406 U.S. 518, 530 (1972) ("The direction of Art. I is that Congress shall have the power to promote the progress of science and the useful arts. When, as here, the Constitution is permissive, the sign of how far Congress has chosen to go can come only from Congress."); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 46 (D. Mass. 1990) ("Under this constitutional mandate, Congress has broad though not unlimited authority to grant copyright monopolies as needed to promote progress.").

right laws all works that would be copyrightable.³⁸ With the Copyright Act of 1909,³⁹ Congress extended copyright protection generally to "all the writings of an author,"⁴⁰ and simply gave examples of works that would be copyrightable.⁴¹

The Copyright Act of 1976⁴² replaced the 1909 Act and expanded the subject matter of copyright as follows: "Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."⁴³ Congress provided a list of the categories of works that could be copyrighted,⁴⁴ but the list was merely intended to provide examples of copyrightable works and was not meant to be exhaustive.⁴⁵ By way of limitation, however, the 1976 Act made it clear that "[i]n no case does copyright protection . . . 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."⁴⁶

38. See *Paperback*, 740 F. Supp at 47 (providing a chronology of additions to the statutory list). Originally, copyright protected maps, charts, and books; by 1870, protection had been extended to, inter alia, etchings and engravings, musical and dramatic compositions, photographs and negatives, and statuary. See sources cited at id.

39. Act of Mar. 4, 1909, ch. 320, 35 Stat. 1075.

40. Id. § 4, 35 Stat. at 1076.

41. See id. § 5, 35 Stat. at 1076-77. The original list included "Books, including composite and cyclopaedic works, directories, gazetteers, and other compilations;" "Periodicals, including newspapers;" "Lectures, sermons, addresses, prepared for oral delivery;" "Dramatic or dramatico-musical compositions;" "Musical compositions;" "Maps;" "Works of art; models or designs for works of art;" "Reproductions of a work of art;" "Drawing or plastic works of a scientific or technical character;" "Photographs;" and "Prints and pictorial illustrations." Motion pictures and sound recordings were added to the list, in 1912 and 1972, respectively.

42. Pub. L. No. 94-553, 90 Stat. 2541 (1976) (codified in 17 U.S.C.).

43. 17 U.S.C. § 102(a) (1988).

44. The list included "literary works," "musical works, including any accompanying words," "dramatic works, including any accompanying music," "pantomimes and choreographic works," "pictorial, graphic, and sculptural works," "motion pictures and other audiovisual works," and "sound recordings." Id. § 102(a)(1)-(7).

The categories "literary works" and "audiovisual works" are particularly significant. Congress intended computer programs to be protected under the former category, see *infra* note 48 and accompanying text, but many people believe that nonliteral elements such as the user interface ought to be protected, if at all, under the latter category, see *infra* text accompanying notes 359-360. But cf. *infra* text accompanying notes 361-364.

45. See H.R. Rep. No. 1476, 94th Cong., 2d Sess. 53 (1976), reprinted in 1976 U.S.C.A.N. 5659, 5666 (stating that list was to be "illustrative and not limitative," and . . . [the] categories do not necessarily exhaust the scope of 'original works of authorship' that the bill [was] intended to protect").

46. 17 U.S.C. § 102(b) (1988).

Although the list of copyrightable subject matter in the 1976 Act did not include computer programs,⁴⁷ the legislative history clearly indicates Congress' intent to protect computer programs under the category of "literary works."⁴⁸ Prior to the passing of the 1976 Act, Congress had established the National Commission on New Technological Uses of Copyrighted Works (CONTU)⁴⁹ to assess the adequacy of the copyright laws in light of modern technological advancements such as photocopy machines and, more importantly, computers.⁵⁰ Congress decided to wait for CONTU's recommendations before taking any major action on computer programs.⁵¹ For the interim period, Congress included a provision in the 1976 Act, § 117,⁵² to "preserve the status quo"—whatever that may have been.⁵³

47. In fact, the language of § 102(b) of the Act, which excludes "process[es], system[s], [and] method[s] of operation" might seem to exclude computer programs from copyright protection altogether.

48. See 17 U.S.C. § 101 (1988) (defining "[l]iterary [w]orks" to include works "expressed in . . . numbers, or other . . . numerical symbols or indicia, regardless of the nature of the material objects, such as . . . disks, . . . in which they are embodied"); see also H.R. Rep. No. 1476, *supra* note 45, at 54, reprinted in 1976 U.S.C.A.N. at 5667 (stating that category of "literary works" includes "computer programs to the extent that they incorporate . . . the programmer's expression of original ideas").

49. See Act of Dec. 31, 1974, Pub. L. No. 93-573, 88 Stat. 1873, 1873-75. CONTU was to be a short-lived commission whose sole purpose was to make recommendations for amendment to the Copyright Act of 1976.

50. See Final Report of the National Commission on New Technological Uses of Copyrighted Works 3 (1979) [hereinafter: CONTU Final Report] (defining CONTU's purpose as "assist[ing] the President and Congress in developing a national policy for both protecting the rights of copyright owners and ensuring public access to copyrighted works when they are used in computer and machine duplication systems, bearing in mind the public . . . interest.").

51. See *id.* at 7 ("In anticipation of the work of [CONTU] and this report, the drafters of the [1976 Act] explicitly stated that it did not address or deal with computer issues.").

52. The original § 117 for the 1976 Act, Pub. L. No. 94-553, § 117, 90 Stat. 2541, 2565 (1976), amended by Pub. L. No. 96-517, § 10(b), 94 Stat. 3082 (1980), reads as follows:

Notwithstanding the provisions of sections 106 through 116 and 118, this title does not afford to the owner of copyright in a work any greater or lesser rights with respect to the use of the work in conjunction with automatic systems capable of storing, processing, retrieving, or transferring information, or in conjunction with any similar device, machine, or process, than those afforded to works under the law, whether Title 17 or the common law or statutes of a State, in effect on December 31, 1977, as held applicable and construed by a court in an action brought under this title.

53. See H.R. Rep. No. 1476, *supra* note 45, at 116, reprinted in 1976 U.S.C.A.N. at 5731.

It was never clear exactly what the status quo was, but since § 117 was to be only an interim measure, it was not a very important issue. See 1 Scott, *supra* note 17, § 3.03, at 3-14 to -16. However, it was clear that this section "was not intended to deny copyright protection to computer programs, but to allow for a period of further study." Raymond T. Nimmer, *supra* note 30, ¶ 1.03[2], at 1-13. Apparently Congress intended to apply protection to computer programs under roughly the same basis as other literary works under the 1909 Act.

The CONTU report⁵⁴ was completed in 1978. CONTU concluded that, since it is very difficult to develop computer programs but quite simple to copy them, copyright protection should be available for computer programs to encourage their creation.⁵⁵ CONTU did not believe that very many changes would be required to achieve its objectives, since it understood copyright law as it existed to be consistent with its views.⁵⁶

CONTU did suggest a few minor changes to the 1976 Act. For example, CONTU recommended that a definition of "computer program"⁵⁷ be added to § 101 of the 1976 Act "to make it explicit that computer programs, to the extent that they embody an author's original creation, are proper subject matter of copyright."⁵⁸ CONTU also suggested that § 117 be repealed "to prevent any question concerning the impropriety of program piracy and to assure that all works of authorship are treated comparably under the new law."⁵⁹ Finally, CONTU advised that a new § 117 be enacted⁶⁰ "to ensure that rightful possessors of copies of computer programs may use or adapt these copies for their use."⁶¹ However, CONTU did not make any recommendations concerning the scope of copyright protection, believing that existing copyright principles were adequate.⁶²

54. CONTU Final Report, *supra* note 50.

55. See *id.* at 11. Computer programs were already implicitly protected by the 1976 Act. See *supra* note 48 and accompanying text.

56. See CONTU Final Report, *supra* note 50, at 12.

57. See *supra* note 13.

58. CONTU Final Report, *supra* note 50, at 1.

59. *Id.* at 12-13.

60. *Id.* at 12. The text of the proposed new section was as follows:

§ 117. *Limitations on exclusive rights: computer programs*

Notwithstanding the provisions of § 106, it is not an infringement for the rightful possessor of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

The proposal was adopted with only a minor change: the word "possessor" was replaced by the word "owner." Cf. 17 U.S.C. § 117 (1988). For the reasons behind this change, see 1 Scott, *supra* note 17, § 3.05[D], at 3-26 to -27.

61. CONTU Final Report, *supra* note 50, at 1.

62. See *id.* at 18 ("[CONTU's] discussion of what rights copyright proprietors have and how those rights are limited does not depend upon [CONTU's] proposal but is based upon various currently existing copyright doctrines.").

Congress adopted CONTU's recommendations almost verbatim in the 1980 Amendments to the 1976 Act.⁶³ For this reason, many courts have considered the CONTU Final Report as a form of surrogate legislative history.⁶⁴ However, such use has more recently drawn criticism, especially with regard to the sections of the 1976 Act that remained unchanged.⁶⁵ Since the CONTU Final Report is filled with ambiguous language,⁶⁶ it is often cited to support contradictory propositions.⁶⁷ Its usefulness is thus questionable, and it should only be cited carefully, and with appreciation of its limitations.⁶⁸

2. *General Principles of Copyright Law.* — The most basic principle of copyright law, known as the idea-expression dichotomy, is that copyright protection extends only to the expression of an idea, and never to the

63. Act of Dec. 12, 1980, Pub. L. No. 96-517, § 117, 94 Stat. 3015, 3028 (1980). However, some minor changes were made to the CONTU recommendations. See, e.g., *supra* note 60.

64. See, e.g., *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1248 (3d Cir. 1983), cert. dismissed, 464 U.S. 1033 (1984); *Williams Elecs., Inc. v. Artic Int'l, Inc.*, 685 F.2d 870, 875-77 (3d Cir. 1982).

65. See, e.g., *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1242 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987) ("There is no sense in which [the CONTU Final Report] represents the will of Congress with respect to provisions not amended in response to the Report."); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 54 (D. Mass. 1990) ("[C]ourts must not treat the CONTU [Final Report] as legislative history, in the ordinary sense, much less as an authoritative statement about manifested legislative intent.").

66. For example, the Report states that "one is always free to make the [computer] do the same thing as it would if it had the copyrighted work placed in it, but only by one's own creative effort rather than by piracy." CONTU Final Report, *supra* note 50, at 21. This could seemingly stand for the proposition that no nonliteral aspects of a computer program should be protected—i.e., that one could make a program that functions identically as long as the program itself is entirely different. On the other hand, it could just as easily stand for the ordinary copyright principle that two authors, working independently, could create identical works which would both be copyrightable. See 3 Nimmer on Copyright, *supra* note 16, § 13.01[B], at 13-13 to -15 ("[T]he trier of fact may be upheld in finding no copying if such trier believes the defendant's evidence of independent creation.").

67. An example can be found in the Report's statement that "[f]low charts, source codes, and object codes are works of authorship in which copyright subsists . . ." CONTU Final Report, *supra* note 50, at 21. This has been interpreted by at least one court to extend copyright protection to the "structure, sequence, and organization" of computer programs. See *Whelan*, 797 F.2d at 1241 ("[T]he reference to the copyrightability of flowcharts [] demonstrates that [CONTU] intended copyright protection to extend beyond the literal code."). On the other hand, it has also been used by at least one commentator to support the proposition that CONTU "drew the line at the literal words of the program." Ramos, *supra* note 32, at 271; see also *id.* at 274 ("CONTU's definition of a flowchart clearly rejects [Whelan's] conclusion.").

68. It should be kept in mind that CONTU made no recommendations regarding the scope of copyright protection, see *supra* note 62 and accompanying text. Moreover, CONTU recognized its own limits and recommended that "[a]ny legislation . . . should be subject to a periodic review to determine its adequacy in the light of continuing technological change." CONTU Final Report, *supra* note 50, at 2.

idea itself.⁶⁹ A copyright confers only the right to the original expression in a given work; the right to an idea (or design or process) is the subject matter of patent law, which has much stricter regulations than does copyright.⁷⁰ Since most progress is evolutionary rather than revolutionary, the rate of advance would be severely impeded if a copyright prevented future authors from using any ideas from previous works.⁷¹ By protecting only an author's expression from unauthorized copying, copyright law strikes a balance that is intended to "promote progress."

Thus, the nonliteral elements of computer programs should receive copyright protection if, but only if, they can be classified as expression rather than as ideas. Yet the distinction is not easy to realize in practice.⁷² Therefore, the courts have developed a number of principles and tests to help them decide whether a particular element of a copyrighted work is an idea or an expression of that idea.

The first of these is the concept of "merger." In the landmark case of *Baker v. Selden*,⁷³ the Supreme Court held that a copyright does not protect processes or methods of operation.⁷⁴ In *Baker*, the plaintiff had written a book describing a system of bookkeeping; the defendant published his own accounting books employing that system. The court found that, although the plaintiff's copyright protected his book from being copied, it did not give him an exclusive right to the "art"—i.e., the bookkeeping system—described; such protection could only be derived from a patent.⁷⁵ Copyright protection for the plaintiff's work extended only to *his description* of the art, which was not copied.⁷⁶ The Court also held that the defendant did not infringe the plaintiff's copyright by using ledger sheets similar to those developed by the plaintiff. The Court reasoned that, since the sheets were necessary to that method of bookkeeping, protecting the sheets would amount to the protection of the system of book-

69. See *Mazer v. Stein*, 347 U.S. 201, 217 (1954) ("[Copyright] protection is given only to the expression of the idea—not the idea itself."); see also 3 Nimmer on Copyright, *supra* note 16, § 13.03[B][2][a], at 13-61 to -65.

70. "[P]atent registration, with its exacting up-front novelty and non-obviousness requirements, might be the more appropriate rubric of protection for intellectual property of this kind." *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992) (citation omitted).

71. "[T]he copyright law has always recognized and tried to accommodate the fact that all intellectual pioneers build on the work of their predecessors." *Whelan*, 797 F.2d at 1238; see also *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 77 (D. Mass. 1990) ("[The] general contention—that 'Progress of Science and useful Arts' cannot occur unless authors . . . are privileged to build upon earlier progress and . . . innovation—has long been a virtually unchallenged premise in all branches of the law of intellectual property.").

72. Judge Learned Hand believed that "[n]obody has ever been able to fix th[e] boundary [between idea and expression], and nobody ever can." *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930).

73. 101 U.S. 99 (1879).

74. This doctrine was subsequently codified into the Copyright law. See 17 U.S.C. § 102(b) (1988), reprinted at *supra* text accompanying note 46.

75. See *Baker*, 101 U.S. at 102-03.

76. See *id.* at 104-05.

keeping itself.⁷⁷ Since the system was not protected, neither could the sheets be protected. This holding became the foundation for the concept of merger: when there is only one way, or a very few ways, to express an idea, the expression is said to merge with its idea and is not protected by copyright.⁷⁸

A related principle is the concept of "*scenes a faire*." *Scenes a faire* are "incidents, characters or settings which are as a practical matter indispensable, or at least standard, in the treatment of a given topic."⁷⁹ For example, one could hardly present Nazi Germany without employing the swastika;⁸⁰ or, in the computer context, one could hardly use symbols other than "+," "-", "*", "/", and "=" to perform mathematical functions in a computer program.⁸¹ Since *scenes a faire* are "indispensable, or at least standard," they are not protected by copyright.⁸² They would otherwise provide back-door copyright protection of ideas to the first person who had used them: because no one else could use the "indispensable" or "standard" expressions of those ideas, it would be difficult to use the ideas at all.

A third limiting doctrine of copyright is the "useful article" doctrine. A "useful article" is defined in the copyright law as "an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information."⁸³ Examples might be the QWERTY keyboard arrangement or the figure-H stick-shift of an automobile.⁸⁴ Under the "useful article" doctrine, a copyright does not extend

77. See *id.* at 103 ("[W]here the art . . . cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public . . .").

78. See *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678 (1st Cir. 1967) ("When the . . . subject matter is very narrow, so that 'the topic necessarily requires,' . . . at best only a limited number [of possible expressions], to permit copyrighting would mean that a party . . . by copyrighting a mere handful of forms, could exhaust all possibilities of future use of the substance."); see also *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971) ("When the 'idea' and its 'expression' are . . . inseparable, copying the 'expression' will not be barred, since protecting the 'expression' in such circumstances would confer a monopoly of the 'idea' upon the copyright owner . . .").

79. *Hoehling v. Universal City Studios, Inc.*, 618 F.2d 972, 979 (2d Cir. 1980) (quoting *Alexander v. Haley*, 460 F. Supp. 40, 45 (S.D.N.Y. 1978)).

80. See *Hoehling*, 618 F.2d at 979.

81. See *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 66-67 (D. Mass. 1990).

82. See *Hoehling*, 618 F.2d at 979 ("Because it is virtually impossible to write about a particular historical era or fictional theme without employing certain 'stock' or standard literary devices, we have held that *scenes a faire* are not copyrightable as a matter of law.").

83. 17 U.S.C. § 101 (1988).

84. The "QWERTY keyboard arrangement" is the standard arrangement of characters on the keyboards of most typewriters and computers. It is named "QWERTY" after the arrangement of the first six letter keys.

The "figure-H stick-shift" is the basic arrangement of gears in the manual transmission of an automobile.

protection to the utilitarian or functional aspects of a work.⁸⁵ However, the mere fact that an article may be useful does not bar protection for any original expression separable from its usefulness.⁸⁶

The courts also employ a test known as the "Abstractions Test" to help them parse an idea from the expression of an idea. Judge Learned Hand first developed this test in *Nichols v. Universal Pictures Corp.*⁸⁷ While that case involved copyright protection for a play, the Abstractions Test has come to be applied generally to all types of literary works. Judge Hand stated the test as follows:

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.⁸⁸

By focusing on a scale of abstraction to distinguish ideas from expressions, this test "implicitly recognizes that any given work may consist of a mixture of numerous ideas and expressions."⁸⁹ The Abstractions Test, however, is not a bright-line test. Judge Learned Hand himself admitted that "[d]ecisions [would] inevitably be *ad hoc*."⁹⁰ Ultimately, it is the task of the decision-maker to weigh all the facts and determine the appropri-

85. "[C]opyrightee [does not have] any rights to the mechanical or utilitarian uses of a work of art." *Rosenthal v. Stein*, 205 F.2d 633, 636 (9th Cir. 1953) (quoting *Stein v. Mazer*, 204 F.2d 472, 477 (4th Cir. 1953)); see also H.R. Rep. No. 1476, *supra* note 45, 54-55, reprinted in 1976 U.S.C.C.A.N. at 5667-68.

86. See *Mazer v. Stein*, 347 U.S. 201, 218 (1954) ("We find nothing in the copyright statute to support the argument that the intended use or use in industry of an article eligible for copyright bars or invalidates its registration."); see also *Rosenthal*, 205 F.2d at 635-37 (arguing that copyrighted work does not lose its status as such by being put to a functional use); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 57-58 (D. Mass. 1990) ("[T]he mere fact that an intellectual work is useful or functional . . . does not mean that none of the elements of the work can be copyrightable.").

87. 45 F.2d 119 (2d Cir. 1930).

88. *Id.* at 121. In applying the Abstractions Test, the court focused on the characters and the "sequence of incident," or the story line, of the plays in question. As to the former, the court noticed some similarities between a few characters, but mostly on the level of their prototypes. This was held to be too abstract to be protected. The court also noted that "the less developed the characters, the less they can be copyrighted." *Id.* As to the latter, the court found similarities only in the broad outline (which was also too abstract to be protected), and generally found the stories to be "quite different." See *id.* at 121. Thus, while there were similarities on the highest levels of abstraction, the details were sufficiently different that the expression in the first play was held not to have been copied by the second play. At best, ideas were common to the two plays, but the ideas were not copyrightable.

89. 3 *Nimmer on Copyright*, *supra* note 16, § 13.03[F][1], at 13-102.18.

90. *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960).

ate level at which to afford copyright protection to any given work.⁹¹ Nevertheless, the Abstractions Test provides a useful framework through which the decision-maker can better understand his or her responsibility.

The idea-expression dichotomy can also be found in the doctrine of "compilations." It is possible even for works consisting entirely of otherwise uncopyrightable elements to be protected under this rubric.⁹² The 1976 Act defines "compilations" as "work[s] 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."⁹³ The Act explicitly limits the protection of compilations by providing that the copyright in such a work protects only the author's original contribution to the compilation⁹⁴—i.e., the selection, coordination, or arrangement. The protection afforded to compilations is "thin".⁹⁵ the Act specifically provides that copyright protection does not extend to "the preexisting material employed in the work, and does not imply any exclusive right in the preexisting material."⁹⁶ Thus, protection for compilations does not contradict the idea-expression dichotomy:⁹⁷ the "ideas" (the otherwise uncopyrightable elements) are not protected while the "expressions" (the selection, coordination, and arrangement of those elements) are.⁹⁸

A related doctrine is that of "total concept and feel,"⁹⁹ originated by the Ninth Circuit Court of Appeals in *Roth Greeting Cards v. United Card Co.*¹⁰⁰ The case involved alleged infringement of a copyright in greeting cards. The court held that, although the textual materials in the greeting cards at issue were not themselves copyrightable because they were in the public domain, a "proper analysis of the problem require[d] that all elements of each card, including text, arrangement of text, art work, and association between art work and text, be considered as a whole."¹⁰¹ The

91. See *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 60–61 (D. Mass. 1990).

92. See *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 111 S. Ct. 1282, 1287–89 (1991) ("[E]ven a directory that contains absolutely no protectible written expression, only facts, meets the constitutional minimum for copyright protection if it features an original selection or arrangement.").

93. 17 U.S.C. § 101 (1988).

94. See 17 U.S.C. § 103(b) (1988) (extending protection "only to the material contributed by the author of such work").

95. See *Feist*, 111 S. Ct. at 1289; see also *infra* text accompanying notes 337–338.

96. 17 U.S.C. § 103(b) (1988).

97. In compilations of facts, the dichotomy is referred to as the "fact/expression dichotomy," but the concept is the same in either case. See *Feist*, 111 S. Ct. at 1290.

98. See *id.*

99. See generally 3 Nimmer on Copyright, *supra* note 16, § 13.03[A][1][c], at 13–37 to –41; see also 2 Howard B. Abrams, *The Law of Copyright* § 14.03[G], at 14–45 to –49 (Release No. 2 1993) [hereinafter Abrams on Copyright].

"Total concept and feel" is also referred to as "look and feel," particularly in the computer program context.

100. 429 F.2d 1106, 1110 (9th Cir. 1970).

101. *Id.* at 1109.

court concluded that each of the plaintiff's cards, considered as a whole, represented expression by the plaintiff and that the defendant had copied the "total concept and feel" of those cards.¹⁰² The *Roth* case thus stands for the proposition that a work can infringe on the copyright of another if it copies the "total concept and feel" of the latter. Unfortunately, through broad usage and expansion, the phrase "total concept and feel" has caused a great deal of confusion: "it is [no longer] clear whether the expression refers to the material to be protected by copyright, to the scope of the copyright protection, or to a test for determining copyright infringement."¹⁰³ The doctrine is therefore highly controversial.¹⁰⁴

Once it has been shown that a particular item is a copyrightable expression and not an uncopyrightable idea, there are certain elements that must be proven to make out a case of copyright infringement. A plaintiff must prove ownership of a valid copyright and copying by the defendant of protected expression.¹⁰⁵ Since actual copying is often difficult to prove, it may be shown indirectly by proving both access by the defendant to the copyrighted work and a substantial similarity of the allegedly infringing work to the copyrighted work.¹⁰⁶ Proving access to a computer program is usually not difficult, especially if it is a widely available commercial product.¹⁰⁷ Thus, the issue in most computer program copyright cases boils down to the question of substantial similarity between those elements of the two programs that are considered to be expressive.

3. *The Early Case Law Establishing the Copyrightability of Computer Programs.* — Two early cases from the Third Circuit, *Williams Electronics, Inc.*

102. See *id.* at 1109–11.

103. Vergari & Shue, *supra* note 18, at 586–92 (discussing "'Look and Feel' cases"). The *Roth* case treated "total concept and feel" as the scope of copyright protection. For a treatment of "total concept and feel" as the material to be protected by copyright, see 1 Scott, *supra* note 17, § 3.47[C][1], at 3–199 ("Courts have found copyright infringement due to the copying of the 'look and feel' [defined as the user interface] of a program in many cases."). For a treatment of "total concept and feel" as a test for determining copyright infringement, see *Litchfield v. Spielberg*, 736 F.2d 1352, 1357 (9th Cir. 1984) ("To constitute infringement of expression, the total concept and feel of the works must be substantially similar.").

104. See 2 Abrams on Copyright, *supra* note 99, § 14.03[G], at 14–45 to –49 ("[T]he phrase 'total concept and feel' should not be used as a test for infringement."); 3 Nimmer on Copyright, *supra* note 16, § 13.03[A][1][c], at 13–37 to –41 ("[T]he frequent invocations of [the 'total concept and feel'] standard do little to bring order to the inquiry into what constitutes substantial similarity, and would be better abandoned."); see also *Lotus Dev. Corp. v. Paperback Software Int'l, Inc.*, 740 F. Supp. 37, 62–63 (D. Mass. 1990) (rejecting "look and feel" as "[not] significantly helpful" in determining copyrightability, and as "conclus[ory]" in that regard).

105. See *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1162 (9th Cir. 1977); see also 3 Nimmer on Copyright, *supra* note 16, § 13.01, at 13–5.

106. See *Sid & Marty Krofft*, 562 F.2d at 1162; see also 3 Nimmer on Copyright, *supra* note 16, § 13.01[B], at 13–11.

107. See 1 Scott, *supra* note 17, § 3.47[A], at 3–192.

*v. Artic International, Inc.*¹⁰⁸ and *Apple Computer, Inc. v. Franklin Computer Corp.*,¹⁰⁹ were particularly helpful in establishing the fundamental copyrightability of the literal elements of computer programs. After these early cases, it was clear that the literal elements of all types of computer programs, whether applications or operating systems,¹¹⁰ were copyrightable in any format, whether source code or object code.¹¹¹

In *Williams*, the Third Circuit confronted, *inter alia*, the question of whether the object code¹¹² of a program was copyrightable.¹¹³ Both parties were developers of video games; the defendant had copied the object code from plaintiff's "Defender" video game. The defendant argued that "a 'copy' must be intelligible to human beings and must be intended as a medium of communication to human beings."¹¹⁴ The court rejected this argument because Congress intended the copyright law to encompass "copies" made by electronic devices. Thus, the court held that computer programs are copyrightable in both source code and object code states.¹¹⁵

In *Apple*, the Third Circuit faced the question of whether the operating system¹¹⁶ of a computer system was copyrightable.¹¹⁷ In addition to expressly affirming a number of holdings from the *Williams* case,¹¹⁸ the court rejected defendant's assertions "that an operating system program is a 'process,' 'system,' or 'method of operation' and hence uncopyrightable."¹¹⁹ The court noted that there was no intrinsic difference between the instructions in an application, which were clearly copyrightable, and

108. 685 F.2d 870 (3d Cir. 1982).

109. 714 F.2d 1240 (3d Cir. 1983).

110. See 1 Scott, *supra* note 17, § 3.11[C], at 3-51 & n.201 (citing cases).

111. See *id.* § 3.11[B], at 3-46 & n.188 (citing cases).

112. For the distinction between "source code" and "object code," see *supra* text accompanying notes 15-16.

113. See *Williams Electronics, Inc. v. Artic Int'l, Inc.*, 685 F.2d 870, 876-77 (3d Cir. 1982).

114. *Id.* at 876-77. This argument is similar to the one which prevailed in *White-Smith Music Publishing Co. v. Apollo Co.*, 209 U.S. 1 (1908). In that case, it was held that a player piano roll did not infringe the copyright on the music it would play because an infringing copy must be "in a form which [people] can see and read." *Id.* at 17. This doctrine was overruled in the 1976 Act by the language in 17 U.S.C. § 102(a) (1988) ("Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."). See 1 Nimmer on Copyright, *supra* note 16, § 2.03[B][1], at 2-28 to -29.

115. See *Williams*, 685 F.2d at 876-77.

116. For the distinction between an "application" and an "operating system," see *supra* text accompanying notes 17-18.

117. See *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1245, 1249-54 (3d Cir. 1983).

118. See *id.* at 1247-49.

119. *Id.* at 1250-51.

those in an operating system. Thus, the court upheld the copyrightability of all types of programs, be they applications or operating systems.¹²⁰

II. A REVIEW OF FOUR SUBSTANTIAL SIMILARITY TESTS FOR DETERMINING THE COPYRIGHTABILITY OF NONLITERAL ELEMENTS OF COMPUTER PROGRAMS

The courts have not been as unanimous about the appropriate scope of protection that should be accorded the nonliteral elements of computer programs as they have been for the literal elements. This Part reviews the four leading cases on the issue of the copyrightability of the nonliteral elements of computer programs. It examines the test for substantial similarity that is provided in each case and reviews some of the criticism that has been leveled against each test.

A. *The Whelan Approach: A Computer Program's "Idea" Is Its Purpose or Function*

In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*,¹²¹ the Third Circuit became the first Circuit Court of Appeals to grapple with the issue of whether the underlying structure of a computer program was copyrightable.¹²² The facts of *Whelan* are as follows.¹²³ Jaslow Dental Laboratories (the "Lab") hired Whelan Associates ("Whelan") to develop a computer program—the "Dentalab" program—to computerize the Lab. The agreement included a provision whereby the developer could sell the program to others and the Lab would receive a ten percent commission on such sales. However, the Lab had been developing its own version of Dentalab, called "Dentcom," which would run on smaller computers and which was therefore expected to have a larger potential market. The Lab and Whelan eventually terminated their agreement, and each party considered itself the rightful owner of the Dentalab program. The Lab then marketed both the Dentalab and the Dentcom programs, and Whelan Associates continued to market the Dentalab program.

In the ensuing litigation, the district court ruled that Whelan was the rightful owner of the copyright to the Dentalab program and that the Lab had infringed this copyright in its sale of both the Dentalab and the Dentcom programs.¹²⁴ The Dentcom program did not copy either the source code or the object code of the Dentalab program—in fact, it was

120. See *id.* at 1251 ("There is . . . no reason to afford any less copyright protection to the instructions in an operating system program than to the instructions in an application program.").

121. 797 F.2d 1222 (3d Cir. 1986).

122. See *id.* at 1224 (stating the issue as "whether the structure (or sequence and organization) of a computer program is protectable by copyright, or whether the protection of the copyright law extends only as far as the literal computer code" (brackets in original)).

123. See *id.* at 1225–27.

124. See *id.* at 1228–29.

written in a different language for a different computer system. Thus, there was no literal copying of the program. However, most of the file structures¹²⁵ and screen outputs of the programs were virtually identical. In addition, five key subroutines performed virtually identically, and there were overall structural similarities.¹²⁶ On appeal, the only issue before the court was whether the Dentcom program infringed upon the copyright of the Dentalab system by copying nonliteral elements of the program.¹²⁷

The circuit court first addressed whether these nonliteral elements were copyrightable. It began by noting that the copyrights of other literary works could be infringed without any copying of the literal elements of the work.¹²⁸ Therefore, the court reasoned, the same should be true of computer programs.¹²⁹ After reiterating the distinction between ideas and expression,¹³⁰ the court developed a test, purportedly based on *Baker v. Selden*,¹³¹ for distinguishing idea from expression in computer programs:

[T]he line between idea and expression may be drawn with reference to the end sought to be achieved by the work in question. In other words, *the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea . . .* Where there are various means of achieving the desired purpose, . . . the particular means chosen is not necessary to the purpose; hence, there is expression, not idea.¹³²

The court went on to apply this principle very broadly:

[T]he purpose of the . . . Dentalab program was to aid in the business operations of a dental laboratory It is . . . clear that the structure of the [Dentalab] program was not essential to that task: there are other programs on the market . . . that perform the same functions but have different structures The conclusion is thus inescapable that the detailed structure of the Dentalab program is part of the expression, not the idea, of that program.¹³³

125. Computers store and retrieve information in "files." A "file structure" is the format in which such information is organized in a file.

126. See 797 F.2d at 1228, 1246-48 (discussing testimony of expert witness as to similarity of programs).

127. See *id.* at 1229.

128. See *id.* at 1234. The court gave this example: "One can violate the copyright of a play or book by copying its plot or plot devices." *Id.* For a discussion of how a work may be copied without literal similarities, see *infra* text accompanying notes 346-349.

129. See 797 F.2d at 1234.

130. See *id.*

131. 101 U.S. 99 (1879); see also *supra* notes 73-78 and accompanying text.

132. 797 F.2d at 1236.

133. *Id.* at 1238-39.

The court believed that public policy—the goal of “promoting progress”—compelled its decision¹³⁴ and concluded that the CONTU Final Report, in its reference to the copyrightability of flow charts,¹³⁵ supported this approach.¹³⁶

The court upheld the lower court’s findings of substantial similarity and announced a new rule that would extend copyright protection beyond a program’s mere source code and object code to its nonliteral elements. In particular, protection would now encompass the “structure, sequence and organization” of computer programs.¹³⁷

The *Whelan* decision is most often criticized as “simplistic” in that “it assumes that only one ‘idea,’ in copyright law terms, underlies any computer program, and that once a separable idea can be identified, everything else must be expression.”¹³⁸ This is unrealistic given the nature of computer programs, with their complex structure of subroutines,¹³⁹ each of which may have its own separable idea.¹⁴⁰

Moreover, this broad definition of a program’s idea¹⁴¹ has been criticized as providing too much protection.¹⁴² Since whatever is not “idea” is protectable as “expression,” it has been argued that the court should have at least defined the idea in a much more detailed way in order to limit this effect of its holding.¹⁴³ Otherwise, programmers will have a difficult time in trying to create competing programs that are at all similar to existing ones.

134. See *id.* at 1237 (“The rule proposed here, which allows copyright protection beyond the literal computer code, would provide 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.”).

135. See CONTU Final Report, *supra* note 50, at 21 (“Flow charts, source codes, and object codes are works of authorship in which copyright subsists.”).

136. See 797 F.2d at 1241. But see *supra* note 67 and accompanying text.

137. See *id.* at 1248 (“[C]opyright protection of computer programs may extend beyond the programs’ literal code to their structure, sequence, and organization.”).

138. 3 Nimmer on Copyright, *supra* note 16, § 13.03[F][1], at 12-102.17.

139. See *supra* note 24 and accompanying text.

140. See *Computer Assocs. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2d Cir. 1992) (“[E]ach subroutine is itself a program, and . . . may be said to have its own ‘idea.’”).

141. See *supra* text accompanying notes 132–133.

142. “*Whelan* has fared . . . poorly in the academic community, where its standard . . . has been widely criticized for being conceptually overbroad.” *Altai*, 982 F.2d at 705 and sources cited therein.

143. The Abstractions Test teaches that the broader the definition of the idea, the more the work will be protected. See *supra* text accompanying notes 87–91; see also *Lotus Dev. Corp. v. Paperback Software Int’l*, 740 F. Supp. 37, 62 (D. Mass. 1990) (noting the incentive for plaintiffs to “urge that the court conceive the ‘idea’ in a very generalized sense,” in order to achieve broad copyrightability for their program); Marc T. Kretschmer, Note, Copyright Protection For Software Architecture: Just Say No!, 1988 Colum. Bus. L. Rev. 823, 837–39 (“[I]f the idea is broadly described, then any idea subsumed by the broader idea becomes classified as expression and thereby enjoys copyright protection.”).

Finally, *Whelan* has been criticized for according protection to the "structure, sequence, and organization" of computer programs.¹⁴⁴ Critics of the *Whelan* test argue a program's structure is more akin to idea than expression and therefore should not be protected.¹⁴⁵ These critics believe that protecting the structure of computer programs would go too far and would impede progress in program development.¹⁴⁶

B. *The Paperback Approach: A Three-Part Test to Determine Copyrightability*

A second approach to copyrightability of nonliteral elements can be found in *Lotus Development Corp. v. Paperback Software International*.¹⁴⁷ The issue before the District Court of Massachusetts in this case was whether the user interface¹⁴⁸ of the popular electronic spreadsheet program, Lotus 1-2-3, was copyrightable.

The facts of *Paperback* are as follows.¹⁴⁹ Recognizing the potential market for an electronic spreadsheet program that was more powerful than those available at the time, Lotus developed its own program, "1-2-3," which quickly became the industry standard. At about the same time, another programmer had been working on his own spreadsheet program. In fact, he had completed a great deal of his own program, including the menu hierarchy of the user interface, before he had ever seen 1-2-3 in operation. However, before marketing the product, he decided that to be successful the product would have to be compatible with 1-2-3. Thus, he changed the menu hierarchy to conform to that of 1-2-3 and marketed the resulting program, "VP-Planner," as a 1-2-3 "workalike." Lotus eventually sued, charging that Paperback Software had infringed its copyright in 1-2-3 by copying the user interface of that program.

144. See, e.g., Peter S. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 Stan. L. Rev. 1045, 1084 (1989) ("[T]he *Whelan* court naively reasoned that because a function could be performed in more than one way, its structure, sequence, and organization is expressive and therefore copyrightable.").

145. See, e.g., Steven R. Englund, Note, Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs, 88 Mich. L. Rev. 866, 896 (1990) ("[P]rotecting structure in its entirety runs the risk of protecting the underlying ideas and processes."); Kretschmer, *supra* note 143, at 840 (arguing that structure, sequence and organization "fall[s] on the idea side of the [idea-expression] continuum").

146. See Michael A. Jacobs, Copyright and Compatibility, 30 Jurimetrics J. 91, 103 (1989) (arguing that "if the structure, sequence and organization rules are extended beyond the facts of *Whelan*, . . . development could be . . . chilled"); Peter G. Spivak, Comment, Does Form Follow Function? The Idea/Expression Dichotomy in Copyright Protection of Computer Software, 35 UCLA L. Rev. 723, 747 (1988) ("The *Whelan* court's protection of nonliteral structural similarity . . . is hopelessly overbroad in theory, . . . produces an inefficient result, . . . [and] will severely impede progress in the computer programming field.").

147. 740 F. Supp. 37 (D. Mass. 1990).

148. For a definition of "user interface," see *supra* notes 29-32 and accompanying text.

149. See 740 F. Supp. at 65-66, 68-69.

As the Third Circuit did in *Whelan*,¹⁵⁰ the district court here pointed out that literary works can be infringed without any literal copying,¹⁵¹ and concluded that the same should hold true for computer programs.¹⁵² Despite the defendant's argument that the proper test for copyrightability should be based on a literal-nonliteral distinction, the court believed that both the CONTU Final Report and congressional intent supported an analysis based on the idea-expression distinction.¹⁵³ Thus, the court developed the following three-part test.¹⁵⁴

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.¹⁵⁵

This step was specifically intended to incorporate Judge Hand's Abstractions Test.¹⁵⁶ In applying this step to the facts, the court noted that, although the idea of an electronic spreadsheet is not copyrightable, a particular expression of the idea of an electronic spreadsheet may be.¹⁵⁷

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.¹⁵⁸

This step incorporated the doctrines of merger and *scenes a faire*.¹⁵⁹ In applying the second step, the court rejected certain aspects of the 1-2-3 program as non-copyrightable,¹⁶⁰ including even "[t]he idea for a two-line

150. See *supra* text accompanying notes 128–129.

151. See 740 F. Supp. at 51–52 ("This type of copying of nonliteral expression, if sufficiently extensive, has never been upheld as permissible copying; rather, it has always been viewed as copying of elements of an expression of creative originality."); see also *infra* text accompanying notes 346–349.

152. See 740 F. Supp. at 54.

153. See *id.*

154. This test was also applied in a subsequent case brought by Lotus before the same court against another competitor, Borland, marketing another 1-2-3 workalike, Quattro Pro. See *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992) (*Borland I*); see also *Lotus Dev. Corp. v. Borland Int'l, Inc.*, No. 90-11662-K (D. Mass.): 831 F. Supp. 223 (1993); 831 F. Supp. 202 (1993); 788 F. Supp. 78 (D. Mass. 1992). For the facts of *Borland II*, see *infra* note 362.

155. 740 F. Supp. at 60.

156. See *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930); see also *supra* text accompanying notes 87–91.

157. See 740 F. Supp. at 65.

158. *Id.* at 61.

159. For "merger," see *supra* notes 73–78 and accompanying text; for "*scenes a faire*," see *supra* notes 79–82 and accompanying text.

160. For example, the "rotated 'L' screen display," and the use of the "/" key to "invoke the menu command system," among other things. See 740 F. Supp. at 66–67.

moving cursor menu."¹⁶¹ However, since that "idea" could be expressed in an almost unlimited number of ways, the overall structure of the menu hierarchy was held to be expression.¹⁶²

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."

In addressing this third element of the test for copyrightability, the decisionmaker is measuring "substantiality" not merely on a quantum scale but by a test that is qualitative as well.¹⁶³

This is the stage at which the actual decision of substantial similarity is made. Applying this step, the court concluded that the details of the menu hierarchy were a substantial part of the program.¹⁶⁴ Thus, the court determined that the copyrightability of the user interface was established.¹⁶⁵

Since it was clear that the developers of VP-Planner had copied the user interface of the 1-2-3 program,¹⁶⁶ the court held that copyright infringement had occurred.¹⁶⁷ This was so even though VP-Planner's user interface was slightly different in that it included additional elements.¹⁶⁸

For the most part, criticism of the *Paperback* case concerns the extension of protection to user interfaces rather than the actual test proposed.¹⁶⁹ It has been asserted that affording the user interface protection is not a good way to carry out the constitutional goal of promoting progress. For example, it has been argued that the user interface of a program is a "useful article,"¹⁷⁰ and therefore should not be pro-

161. Id. at 65.

162. See id. at 68.

163. Id. at 61.

164. See id. at 68.

165. See id. at 68-69.

166. See id. ("Not only is the copying in this case so 'overwhelming and pervasive' as to preclude, as a matter of law, any assertion of independent creation, . . . but also, defendants in this case have admitted that they copied these elements of protected expression.").

167. See id. at 70.

168. "[S]ome additional commands are included at the end of some menu lines" Id.

169. See generally Gerard J. Lewis, Jr., Comment, Lotus Development Corp. v. Paperback Software International: Broad Copyright Protection For User Interfaces Ignores the Software Industry's Trend Toward Standardization, 52 U. Pitt. L. Rev. 689, 692-93 (1991) (stressing need for software standardization); Joseph T. Verdesca, Jr., Comment, Copyrighting the User Interface: Too Much Protection?, 45 Sw. L.J. 1047, 1075-80 (1991) (same). But see Ramos, *supra* note 32, at 281-84 (arguing that *Paperback* fails to address process-expression distinction adequately).

170. See *supra* notes 83-86 and accompanying text.

tected by copyright.¹⁷¹ The *Paperback* court explicitly considered and rejected this argument. The court clarified the "useful article" doctrine, stating that while usefulness does not add to a copyright claim, neither does it detract from such a claim: "Elements of expression, even if embodied in useful articles, are copyrightable if capable of identification and recognition independently of the functional ideas that make the article useful."¹⁷²

Another criticism that has been leveled against the result in *Paperback* is that copying the user interface of a de facto standard product is necessary to achieve standardization and compatibility in the software industry.¹⁷³ However, the court considered and rejected this argument as well, stating that "the desire to achieve 'compatibility' or 'standardization' cannot override the rights of authors"¹⁷⁴

One last criticism often made against *Paperback* is that since "innovation occurs by building on previous works," programmers should be free to borrow and improve upon the ideas of other programmers.¹⁷⁵ Again, however, the court rejected such arguments, believing that as long as programmers are permitted to "borrow and improve upon previous ideas—such as the ideas for an electronic spreadsheet and a two-line moving cursor menu . . . [a]dequate room for innovation remains even though successors are barred from copying earlier authors' particular expressions"¹⁷⁶

C. The Brown Bag Approach: The Extrinsic-Intrinsic Test

A third approach to the copyrightability of nonliteral elements of computer programs was set forth in *Brown Bag Software v. Symantec Corp.*¹⁷⁷ The test employed by the Ninth Circuit in that case was not

171. See Lewis, *supra* note 169, at 710–12, 714–18; see also *Synercom Technology, Inc. v. Universal Computing Co.*, 462 F. Supp. 1003, 1013 (N.D. Tex. 1978) (concerning computer input formats).

172. 740 F. Supp. at 58; see also *Rosenthal v. Stein*, 205 F.2d 633, 635–37 (9th Cir. 1953); 740 F. Supp. at 57–58.

The *Paperback* court refused to pronounce a rule that would protect inferior works but deny protection to superior works:

[T]he statute does not bar copyrightability merely because the originality of the expression becomes associated, in the marketplace, with usefulness of the work to a degree and in dimensions not previously achieved by other products on the market Rather than promoting and encouraging both the development and disclosure of the best, such a rule would offer incentives to market only the second, or third, or tenth best, and hold back the best for fear that it is too good for copyrightability. Copyrightability is not a synonym for imperfection.

Id. at 58 (citations omitted).

173. See Menell, *supra* note 144, at 1092–98; Verdesca, *supra* note 169, at 1075–79.

174. 740 F. Supp. at 69.

175. See Ramos, *supra* note 32, at 285.

176. 740 F. Supp. at 78.

177. 960 F.2d 1465 (9th Cir. 1992).

novel;¹⁷⁸ the court was merely applying a general test of copyrightability in the computer program context.¹⁷⁹

The relevant facts of *Brown Bag* are as follows.¹⁸⁰ John L. Friend, an independent computer programmer, developed an outlining program called "PC-Outline." Friend sold the program to Brown Bag Software ("BBS"), but retained a non-exclusive right to part of the source code. Subsequently, Friend developed another outlining program called "Grandview." Friend sold this program to Symantec Corporation ("Symantec") who marketed the program as an updated version of its own outlining programs. BBS sued Symantec and Friend alleging, inter alia, that the Grandview program infringed BBS's copyright in PC-Outline. In response to a summary judgment motion by the defendants, BBS submitted a declaration setting forth seventeen features of the Grandview program that were allegedly similar to those in the PC-Outline program. The district court held that none of these features could support a finding of copyright infringement and therefore granted the defendants' summary judgment motion. BBS appealed.

On appeal, the Ninth Circuit did not address the issue of whether the nonliteral elements of computer programs were protected by the copyright law; existing precedent in the Ninth Circuit had established

178. The test had already been developed by existing case law. It was originally derived from the bifurcated test of *Arnstein v. Porter*, 154 F.2d 464 (2d Cir. 1946). The first part of this test determined whether copying had occurred; here, analytical dissection and expert testimony were permitted. The second part of the test determined whether the copying was illicit; here, only the response of the ordinary lay observer was permitted. See id. at 468-69.

The Ninth Circuit adopted a modified form of the bifurcated test—the Extrinsic-Intrinsic Test—in *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1164-65 (9th Cir. 1977). In the first phase, the "extrinsic test," an objective determination is made as to whether there is a substantial similarity of ideas in the two works; again, analytic dissection and expert testimony were permitted. In the second phase, the "intrinsic test," a subjective determination is made as to whether there is a substantial similarity in expressions of the two works; again, the determination depended upon the response of the ordinary reasonable person. See id. at 1164. Although the *Sid & Marty Krofft* court quoted the Abstractions Test with approval, see id. at 1163, it did not explicitly adopt the Abstractions Test as a part of its Extrinsic-Intrinsic Test. See id. at 1163-65.

The Ninth Circuit further modified the test in the case of *Shaw v. Lindheim*, 919 F.2d 1353 (9th Cir. 1990). The court stated that "[b]ecause the criteria incorporated into the extrinsic test encompass all objective manifestations of creativity, the two tests are more sensibly described as objective [for the extrinsic test] and subjective [for the intrinsic test] analyses of *expression*, having strayed from *Krofft's* division between expression and ideas." Id. at 1357. This is the current form of the test and the one adopted by *Brown Bag* for the computer program context. See 960 F.2d at 1474.

179. Neither *Sid & Marty Krofft* nor *Shaw* involved computer programs. The *Sid & Marty Krofft* case involved the fictional worlds and characters in the "H. R. Pufnstuf" television show and the "McDonaldland" television commercials. See 562 F.2d at 1160. The *Shaw* case involved a television series, "The Equalizer." See 919 F.2d at 1355.

180. See 960 F.2d at 1468-69, 1472-73.

that they were if they qualified as expression.¹⁸¹ Instead, the court simply applied the existing two-part test for determining substantial similarity—the Extrinsic-Intrinsic Test.¹⁸²

The first phase of that test is the “extrinsic test,” which involves an “objective . . . analys[is] of *expression*.”¹⁸³ In this phase, individual elements of the copyrighted work are analytically dissected—and unprotectable elements of expression eliminated from consideration—in order to define the scope of the plaintiff’s copyright.¹⁸⁴ Since it is an objective test, expert testimony is admissible and a determination can be made as a matter of law.¹⁸⁵

In applying the extrinsic test, the district court had compared the list of features, including screen displays, function menus, and keystrokes, that were alleged by the plaintiff to be similar.¹⁸⁶ The district court had analyzed these features and concluded that they could not support a claim of copyright infringement under various copyright doctrines: e.g., some features basically merged with their ideas or were *scenes a faire*, others were purely functional, and still others had been licensed to Mr. Friend by BBS.¹⁸⁷ The court had also held that the remaining features were simply not substantially similar as a matter of law.¹⁸⁸ Thus, the court had granted summary judgment in favor of the defendants. The Ninth Circuit upheld the substance of this analysis in its entirety.¹⁸⁹

The second phase of the Extrinsic-Intrinsic Test is the “intrinsic test.” It is triggered only in cases where the court finds objective similarities of protected expression.¹⁹⁰ The “intrinsic test” is a subjective test that meas-

181. “Whether the non-literal components of a program, including the structure, sequence and organization and user interface, are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself.” *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989) (upholding finding of substantial similarity).

182. For the development of this test in prior case law, see *supra* note 178.

183. 960 F.2d at 1475 (quoting *Shaw*, 919 F.2d at 1357).

184. “[A]nalytic dissection is used not for the purposes of comparing similarities and identifying infringement, but for the purpose of defining the scope of plaintiff’s copyright.” *Id.* at 1475–76.

185. “Since it is an extrinsic test, analytic dissection and expert testimony are appropriate. Moreover, this question may often be decided as a matter of law.” *Sid & Marty Krofft Television Prods., Inc. v. McDonald’s Corp.*, 562 F.2d 1157, 1164 (9th Cir. 1977).

186. See 960 F.2d at 1475.

187. See *id.* at 1472–73.

In this stage of the “extrinsic test,” the court excludes any elements that cannot be protected by the plaintiff’s copyright from consideration. For a more detailed analysis under the “extrinsic test,” see *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1020–22 (N.D. Cal. 1992).

188. See 960 F.2d at 1473.

189. See *id.* (“We reject each of [the plaintiff’s] contentions.”).

190. “If the similarity of the works in suit stems solely from unprotectible features, then the plaintiff’s case is missing an essential element of infringement . . . [and] summary judgment is plainly appropriate before reaching the ‘intrinsic test.’” *Apple*

ures "substantial similarity in expressions . . . depending on the response of the ordinary reasonable person" ¹⁹¹ It compares the protected elements determined under the "extrinsic test" ¹⁹² to assess—without the help of analytic dissection or expert testimony ¹⁹³—whether there is substantial similarity in the "total concept and feel" of the two works. ¹⁹⁴ Since the test is purely subjective, it is "a task no more suitable for a judge than for a jury," and is therefore a question for the trier of fact. ¹⁹⁵ Neither the district court nor the Ninth Circuit ever reached the "intrinsic test" in *Brown Bag* since, after its ruling on the "extrinsic test," there were no remaining triable issues of fact.

The Extrinsic-Intrinsic Test has been criticized on the ground that, if a work fails to pass the "extrinsic test," the "intrinsic test" is never triggered. ¹⁹⁶ Since under the "extrinsic test" the court dissects the program into its elements and analyzes each element's copyrightability individually, the court may never consider whether the program as a whole deserves copyright protection. ¹⁹⁷ Moreover, the distinction between objectivity and subjectivity in analysis of similarities has been characterized as "rather malleable" and easily manipulable. ¹⁹⁸ Thus, it would be

Computer, 799 F. Supp. at 1020 (citations omitted); cf. *Brown Bag*, 960 F.2d at 1476 ("[S]atisfaction of the extrinsic test creates a triable issue of fact . . ." (quoting *Shaw v. Lindheim*, 919 F.2d at 1359 (9th Cir. 1990))).

191. 960 F.2d at 1475 (quoting *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1164 (9th Cir. 1977)).

192. "[T]he 'intrinsic test' entails a comparison of the portions of a work that can be the subject of copyright protection." *Apple Computer*, 799 F. Supp. at 1020.

However, it is not entirely clear what is supposed to be compared in the intrinsic test: The degree to which unprotected or unprotectable features must be eliminated from a comparison of two works is difficult to say. Although copyright protection is not afforded to certain elements of a work, such limitations "must not obscure the general proposition that copyright may inhere, under appropriate circumstances, in the selection and arrangement of unprotected components."

Brown Bag, 960 F.2d at 1476 n.4 (quoting 3 Nimmer on Copyright, supra note 16, § 13.03[F][5]).

193. See 960 F.2d at 1475 (quoting *Sid & Marty Krofft*, 562 F.2d at 1164).

194. See *Shaw*, 919 F.2d at 1360 ("What remains is a subjective assessment of the 'concept and feel' of the two works . . .").

195. *Id.*

196. See, e.g., David A. Lowe, Comment, A Square Peg in a Round Hole: The Proper Substantial Similarity Test for Nonliteral Aspects of Computer Programs, 68 Wash. L. Rev. 351, 363 (1993) (arguing that nature of computer programs is such that while two works may appear dissimilar in literal aspects, "look and feel" may in fact be quite similar).

This was even more problematic before the *Shaw* opinion transformed the "extrinsic test" into an objective analysis of expression, see supra note 178. When the "extrinsic test" was an analysis of similarity of ideas, it was theoretically possible for the court to grant a summary judgment on the ground that the ideas of the two works in question were different, without ever considering the similarities in expression. See Mitchell Zimmerman, Substantial Similarity of Computer Programs After *Brown Bag*, 9 No. 7 Computer Law. 6, 7-8 (1992).

197. See Lowe, supra note 196, at 363.

198. See Zimmerman, supra note 196, at 15.

simple for a judge inclined against protection to characterize the expressions at issue as "objectively" dissimilar and thereby avoid the "intrinsic test" entirely.¹⁹⁹

The "intrinsic test" has itself been faulted as an amorphous test more akin to a gut reaction than to analysis.²⁰⁰ It has been argued that, particularly in cases involving complex subjects such as computer programs, such an oversimplified analysis is entirely inappropriate.²⁰¹ Moreover, it has been criticized as a dangerous test in that it is very likely to result in the protection of ideas themselves rather than just expression.²⁰²

However, the area in which *Brown Bag* is most susceptible to criticism is in the inadequacy of definition of its test. There is no formulation of the test other than at the most general levels. The "analytic dissection" of the "extrinsic test" is a highly complex process,²⁰³ yet the court provides scant guidance as to what it entails other than incidentally by way of example.²⁰⁴ The court also hedges its description of the "intrinsic test" and never definitively states what should be considered during that stage.²⁰⁵ The test needs refinement.

D. *The Altai Approach: The Abstraction-Filtration-Comparison Test*

In *Computer Associates International, Inc. v. Altai, Inc.*,²⁰⁶ the Second Circuit articulated a fourth test for determining whether the nonliteral

199. See *id.* at 15-16.

200. The "intrinsic test" embodies the doctrine of "total concept and feel." See *supra* text accompanying note 194. This doctrine has been criticized extensively. See *supra* note 104.

For a general review of the doctrine of "total concept and feel," see *supra* text accompanying notes 99-104.

201. See, e.g., 3 Nimmer on Copyright, *supra* note 16, § 13.03[A][1][c], at 13-39 to -40 ("The phrase is geared towards simplistic works that require only a highly 'intrinsic' (i.e., unanalytic) evaluation; it serves no purpose in the realm of computers where analytic dissection and expert testimony emphatically are needed." (footnote omitted)).

202. See, e.g., *id.* at 13-40 ("[T]he touchstone of 'total concept and feel' threatens to subvert the very essence of copyright, namely the protection [only] of original *expression*." (footnotes omitted)); see also 2 Abrams on Copyright, *supra* note 99, § 14.05[A][3], at 14-63 ("There is nothing in the definitions of concept and feel that indicate these words are useful tools to sort out copying of expression from copying of ideas. If anything, the opposite is true. The word '[concept]' is a synonym for idea, and not for expression." (footnote omitted)).

203. For a thorough application of the analytic dissection of the "extrinsic test," see *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1027-41 (N.D. Cal. 1992) (comparing visual displays of Apple Macintosh with those of Microsoft Windows).

204. The best guidance comes from the court's description of what the district court had done. See *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1472-73 (9th Cir. 1992). Not much additional guidance can be gleaned from the other decisions that have helped to develop the Extrinsic-Intrinsic Test. See *Shaw v. Lindheim*, 919 F.2d 1353 (9th Cir. 1990); *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157 (9th Cir. 1977). This stands in stark contrast to the other court opinions considered in this Note which have managed adequately to describe their own tests.

205. See *supra* note 192 and accompanying text.

206. 982 F.2d 693 (2d Cir. 1992).

elements of a computer program would be protected by copyright.²⁰⁷ The facts of *Altai* are as follows.²⁰⁸ Computer Associates ("CA") developed "ADAPTER," an "operating system compatibility component"²⁰⁹ within one of its programs that allowed the program to function on various operating systems. One of the developers of the ADAPTER program was subsequently hired by Altai to create a version of its own program for other operating systems. Unbeknownst to Altai, this programmer developed a "common system interface"²¹⁰—to be called "OSCAR 3.4"—based on his familiarity with the ADAPTER program; he even directly copied 30% of ADAPTER's code. When CA sued Altai, Altai had programmers who had not been involved in the development of OSCAR 3.4 rewrite the stolen sections of the code. The new program, "OSCAR 3.5," was included in all new sales and was offered as a "free upgrade" to all previous purchasers.

CA brought suit alleging that Altai had infringed CA's copyright in ADAPTER by copying the code and the structure of the program in its OSCAR 3.4 and OSCAR 3.5 programs. The district court held Altai liable for copying ADAPTER into the OSCAR 3.4 program;²¹¹ this decision was not appealed. The district court also held that Altai was not liable for copyright infringement in developing OSCAR 3.5,²¹² and this decision was appealed by CA.²¹³

In determining whether the nonliteral elements of the program were copyrightable, the Second Circuit noted, as did the previous courts, that copyright protection generally extends beyond a literary work's strictly textual form to its nonliteral elements of expression.²¹⁴ The court therefore looked to the idea-expression dichotomy in deciding the case.²¹⁵ However, the court rejected the *Whelan* approach as simplistic and conceptually overbroad,²¹⁶ announcing instead its own "Abstraction-Filtration-Comparison" test.²¹⁷

207. See *id.* at 706–11.

208. See *id.* at 698–700.

209. An "operating system compatibility component" is a program that allows the main program to be compatible with various operating systems.

210. Basically, a "common system interface" is just another name for an "operating system compatibility component."

211. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 560–61 (E.D.N.Y. 1991).

212. See *id.* at 562.

213. The court also made trade secret rulings which were appealed. See *Computer Assocs., Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 715–21 (2d Cir. 1992); 775 F. Supp. at 562–67. These rulings, however, are not considered in this Note.

214. See 982 F.2d at 701; see also *infra* text accompanying notes 346–349.

215. See 982 F.2d at 703–06.

216. See *id.* at 705–06.

217. See *id.* at 706–11. The court was basically adopting the "successive filtering method" developed in *3 Nimmer on Copyright*, *supra* note 16, § 13.03[F]. See 982 F.2d at 707.

Initially, in 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.²¹⁸

This step incorporates Judge Hand's Abstractions Test.²¹⁹ The court accepted the district court's assessment of the levels of abstraction to include "in order of 'increasing generality' from object code, to source code, to parameter lists,²²⁰ to services required, to general outline."²²¹

The second step, calling on traditional principles of copyright, filters out those elements of the program that are not copyrightable:

This process entails examining the structural components at each level of abstraction to determine whether their particular inclusion at that level was "idea" or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain and hence is nonprotectable expression.²²²

Elements "dictated by considerations of efficiency" are to be excluded from copyright protection under the concept of merger.²²³ Elements "required by factors external to the program itself"—upon which the opinion elaborates²²⁴—are to be excluded under the concept of *scenes a faire*.²²⁵ Finally, elements "taken from the public domain" are not protected because they are not "original works of authorship" under the 1976 Act.²²⁶ Applying this step, the court agreed that there was no similarity in the two programs' codes and in most of the parameter lists and macros,²²⁷ and that the list of services provided and organizational structures were not protected because they were dictated by the nature of the

218. 982 F.2d at 707.

219. See *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930); see also *supra* text accompanying notes 87–91.

220. The court defined "parameter lists" as "the form in which information is passed between [subroutines]." 982 F.2d at 697–98. (footnote added)

221. *Id.* at 714 (quoting *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 560 (E.D.N.Y. 1991)). However, the court noted that "the facts of a different case might require that a district court draw a more particularized blueprint of a program's overall structure." *Id.*

222. *Id.* at 707.

223. See *id.* at 707–09; *supra* notes 73–78 and accompanying text.

224. The court elaborates as follows:

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

982 F.2d at 709–10 (citing 3 *Nimmer on Copyright*, *supra* note 16, § 13.03[F][3]).

225. See *id.* at 707, 709–10; *supra* notes 79–82 and accompanying text.

226. See 17 U.S.C. § 102(a) (1988).

227. "A macro is a single instruction that initiates a sequence of operations . . . within the program." 982 F.2d at 698.

program.²²⁸ The court found that only a few parameter lists and macros could withstand scrutiny and be protected by copyright.²²⁹

The third step of the *Altai* test was expressed as follows:

Once a court has sifted out all elements of the allegedly infringed program which are [non-protectable], there may remain a core of protectable expression At this point, the court's substantial similarity inquiry focuses on whether the defendant copied any aspect of this protected expression, as well as an assessment of the copied portion's relative importance with respect to the plaintiff's overall program.²³⁰

This is the comparison stage, at which the decision-maker determines whether any copying constitutes infringement. The court upheld the district court's finding that since only a few copyrightable elements remained, a finding of copyright infringement was not required given their relative unimportance to the program.²³¹

The *Altai* decision has been greeted with a mixed reception by both the courts²³² and commentators.²³³ The test should have the support of the leading expert in the field, Professor David Nimmer, since the test was based on the "successive filtering method" from his own *Nimmer on Copyright* treatise.²³⁴

However, the *Altai* test has been criticized as being underprotective due to the rigors of the successive filtering.²³⁵ Moreover, another professor, Raymond T. Nimmer, has challenged the *legitimacy* of the holding, apparently believing that it was based on a presumption of what programmers would prefer rather than on a neutral reading of the law.²³⁶ He argues that the "efficiency demands of programmers"—i.e., that they be

228. See *id.* at 714-15.

229. See *id.*

230. *Id.* at 710 (citations omitted).

231. See *id.* at 714-15.

232. See 1 Scott, *supra* note 17, § 3.47[B][5], at 3-198.6 & n.966.16 (citing cases).

233. Compare Dennis M. McCarthy, Recent Decision, Copyright Infringement—Redefining the Scope of Protection Copyright Affords the Non-Literal Elements of a Computer Program—*Computer Associates International, Inc. v. Altai, Inc.*, 66 Temp. L. Rev. 273 (1993) (praising *Altai* for its realistic understanding of computer programming and its promotion of both private innovation and public good) and Timothy S. Teter, Note, Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases, 45 Stan. L. Rev. 1061, 1084-86 (1993) (approving *Altai*'s use of merger doctrine which allows copying of elements needed to achieve compatibility), with Raymond T. Nimmer, *supra* note 30, Special Update (explaining that concerns about efficiency and technological development, rather than wording of statute, drive *Altai* decision) and Recent Case, 106 Harv. L. Rev. 510, 511 (1992) (predicting that *Altai*'s three-part test will "discourage innovative programming techniques and leave non-literal elements of computer programs under-protected").

234. See 982 F.2d at 707; see also 3 Nimmer on Copyright, *supra* note 16, § 13.03[F], at 13-102.9 to -102.35.

235. See, e.g., Recent Case, *supra* note 233, at 512-14 (criticizing use of merger and *scènes à faire* doctrines as problematic in computer program context).

236. See Raymond T. Nimmer, *supra* note 30, Special Update 4-5.

free to copy whatever they find efficient in a competing program—and the “demands of compatibility and interface”—i.e., that programmers be free to copy whatever is required by external factors, such as compatibility—are not valid defenses to copyright infringement on current statutory grounds.²³⁷ Although such “efficiency demands” and “compatibility requirements” may be the appropriate subject of legislation by Congress, they find little or no support in the 1976 Act.²³⁸

This criticism raises some valid concerns. A copyright should not be subject to qualification merely because a competitor finds the copyrighted expression to be the “best” in some sense. Nevertheless, a careful reading of this so-called “right to copy[] based on the demands of [efficiency]”²³⁹ as developed in *Nimmer on Copyright* reveals that the concerns are not as problematic as they seem. It is true that both the Second Circuit and the *Nimmer* treatise list a number of extrinsic considerations that would limit the copyrightability of a program,²⁴⁰ but these are merely applications of copyright principles, such as merger or *scènes a faire*, in the computer program context. Unlike ordinary literary works, where the range of expression is limited only by the author’s imagination, computer programs must be written in a logical format so that they can be understood by the computer:²⁴¹ computers simply cannot understand such expressive literary devices as metaphor, simile, hyperbole, and the like. This necessarily limits, though not entirely, the range of possible expression at least of a program’s literal elements and structure.²⁴² The doctrines of merger and *scènes a faire* are perfectly suited to taking such factors into consideration.

Moreover, none of the extrinsic limitations on copyrightability—e.g., the “compatibility requirements”—are intended to permit illicit copying of “original works of authorship.” Instead, they generally concern either elements that as a practical matter are not likely to be guarded by the original developer²⁴³ or elements that are not “original works of author-

237. *Id.* at 4.

238. See *id.*

239. *Id.* at 5.

240. See *Computer Assocs. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693, 709-10 (2d Cir. 1992); 3 *Nimmer on Copyright*, supra note 16, § 13.03[F][3], at 13-102.21 to -102.28.

241. See supra text accompanying notes 12-13.

242. Since computers can perform only a limited number of operations, individual lines of the program’s literal code will obviously not be very expressive. Combining these operations to create more complex functions and the overall structure of a program clearly allows for more expression. Nevertheless, the range of possible expression is still significantly limited by comparison to natural languages.

243. One such factor not likely to be jealously guarded is “Software Standards” or compatibility requirements. See 3 *Nimmer on Copyright*, supra note 16, § 13.03[F][3][b], at 13-102.23 to -102.24. While the mere fact that designers would like a program to be compatible with another does not necessarily permit copying, see *Lotus Dev. Corp. v. Paperback Software Int’l*, 740 F. Supp. 37, 69 (D. Mass. 1990), *Nimmer* limits his discussion to compatibility with operating systems. These are usually created to encourage compatibility: the more applications that are available for it, the more popular the

ship" at all.²⁴⁴ At no point does this test suggest that otherwise copyrightable elements should be unprotected merely because programmers would find this result beneficial.

III. ANALYSIS AND PROPOSAL

As discussed above, the courts have developed four separate tests for determining the scope of copyright protection that should be afforded to the nonliteral elements of computer programs. This Part compares the four tests to determine to what extent they are compatible and, to the extent that they are not, whether they are reconcilable. Next, an attempt is made to determine which of the tests is preferable from the standpoint of copyright doctrine. After concluding that the best approach would be a modified version of the most sophisticated test, the *Altai* test, an additional step in the analysis is proposed to deal with problems that may be inherent to that test. Finally, some of the major concerns regarding the copyrightability of nonliteral elements are re-evaluated from the perspective of this *modified Altai* test.

A. Comparison and Reconciliation of the Tests

The court that developed each test quoted Judge Hand with praise²⁴⁵—but not every court adopted his Abstractions Test.²⁴⁶ The *Pa-*

operating system becomes. Cf. *Apple Computer, Inc. v. Microsoft Corp.*, 717 F. Supp. 1428, 1431 (N.D. Cal. 1989) ("Both Apple and Microsoft rely heavily on third-party programmers to develop applications programs to run under their respective operating environments, thus enhancing the value of the operating environments.").

Another such factor is "Manufacturers' Design Standards." See 3 Nimmer on Copyright, *supra* note 16, § 13.03[F][3][c], at 13-102.24 to -26. These are not likely to be jealously guarded for the same reasons that operating systems are not. Thus, such factors do not present a problem.

244. One such factor is "Target Industry Practices." See 3 Nimmer on Copyright, *supra* note 16, § 13.03[F][3][d], at 13-102.26. In this case, Nimmer is referring to two programs which may be similar based on factors external to both of them, such as "business practices and technical requirements," and therefore beyond their control. See *id.* Clearly program elements based on such external factors are not "original works of authorship" under the 1976 Act and would not be protected. See 17 U.S.C. § 102(a) (1988).

Another factor is "Computer Industry Programming Practices." See 3 Nimmer on Copyright, *supra* note 16, § 13.03[F][3][e], at 13-102.27 to -102.28. These are obviously not original to the programmer.

Nimmer also lists "Hardware Standards." See *id.* § 13.03[F][3][a], at 13-102.21 to -102.23. In order to make a computer perform a certain function or process, particular lines of code may be necessary. But since it is generally accepted that "one is always free to make a machine perform any conceivable process," CONTU Final Report, *supra* note 50, at 20, any similarity between programs based on the requirements of the computer cannot be protected.

245. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 706 (2d Cir. 1992); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1235 (3d Cir. 1986); *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1163 (9th Cir. 1977) (quoted in *Shaw v. Lindheim*, 919 F.2d 1353, 1356 (9th Cir. 1990)); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 60 (D. Mass. 1990).

246. See *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930); *supra* text accompanying notes 87-91.

perback and *Altai* tests explicitly incorporate the Abstractions Test; the *Whelan* and *Brown Bag*²⁴⁷ tests do not. Thus, the first issue to be considered is whether the Abstractions Test is incompatible with the *Whelan* and *Brown Bag* decisions.

Whelan might at first glance seem incompatible with the Abstractions Test. After all, the Abstractions Test states that the "idea" of a work can be adjudged at various levels of specificity,²⁴⁸ while *Whelan* stated that the idea of a work is its purpose or function.²⁴⁹ Not everyone agrees with this interpretation of *Whelan*, however.²⁵⁰ The *Whelan* court itself stated that the idea of a work will not always be its purpose or function.²⁵¹ As an example, the court stated that "[t]he idea . . . may be to accomplish a certain function *in a certain way*."²⁵² Thus, the court implicitly recognized that the idea of a work may be adjudged at various levels of specificity after all.²⁵³ In the end, all of the approaches face the same question of where to draw the line between idea and expression.²⁵⁴ Thus, the Abstractions Test could to a large extent be incorporated into the *Whelan* test without substantially altering its character: when deciding how to characterize the program's purpose—i.e., its idea—the court could easily apply the Abstractions Test.²⁵⁵

247. The "*Brown Bag* test" will be used hereinafter to refer to the Extrinsic-Intrinsic Test of the Ninth Circuit, see *supra* section II.C.

248. For a review of the Abstractions Test, see 45 F.2d at 121; *supra* text accompanying notes 87–91.

249. See 797 F.2d at 1236; *supra* text accompanying note 132.

250. Professor Raymond T. Nimmer states:

[M]any observers regard *Whelan* as establishing a view that a single idea can describe a program and that this idea can be stated in very general, nonspecific terms. . . . In fact, however, *Whelan* held that copyright infringement occurs only if the details of a complex organization or structure have been copied and only if *those details* are not necessary to the idea of the work.

Raymond T. Nimmer, *supra* note 30, Special Update 1 (emphasis in original).

251. "We do not mean to imply that the idea or purpose behind *every* utilitarian or functional work will be precisely what it accomplishes, and that structure and organization will therefore always be part of the expression of such works." *Whelan*, 797 F.2d at 1238 n.34.

252. *Id.*

253. At one level of abstraction—probably the highest—the idea is "the efficient organization of a dental laboratory." *Id.* at 1240. At another level, the idea might be the efficient organization of a dental laboratory *in a certain way*.

254. Compare *supra* note 253 with *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 216 (D. Mass. 1992) (*Borland II*) (setting forth a number of possible "ideas" for the 1-2-3 program at various levels of abstraction).

255. To a certain extent, incorporating the Abstractions Test into the *Whelan* test would be unavoidable. The *Whelan* court was faced with a single purpose program: one created to manage dental laboratories. More often, however, the program in question is a general purpose program. For instance, *Paperback* dealt with a spreadsheet program. In such a case, it would be difficult to ascribe one particular purpose or function to the program. Regardless of whether a court explicitly adopted the Abstractions Test, it would inevitably face the challenge of deciding at what level of detail to describe the program's "idea."

Under the *Brown Bag* test, on the other hand, the court never has the opportunity to focus on the difference between idea and expression in the program. Instead, the court is presented with a list of elements which the plaintiff claims were copied in the infringing work.²⁵⁶ To be sure, the court must decide whether each element is closer to idea or expression,²⁵⁷ but the benefit of the Abstractions Test is lost: since the court never systematically evaluates the relationship between idea and expression in the program, each individual element is truly evaluated on an entirely ad hoc basis.²⁵⁸

A closely related issue is whether the tests are capable of recognizing that a program can have more than just one idea. Both the *Paperback* and the *Altai* courts do.²⁵⁹ However, the *Altai* court does so more explicitly, recognizing even the existence of multiple ideas *horizontally*—i.e., recognizing that each subroutine, function, or feature of the program can have its own idea.²⁶⁰ The *Paperback* decision only seems to recognize the concept of multiple ideas *vertically*—i.e., that a program's idea can be expressed in various degrees of abstraction.²⁶¹ Nevertheless, there is nothing inherent to the *Paperback* test that prohibits the recognition of multiple ideas horizontally.²⁶² If the first step in the *Paperback* test is permitted to recognize such multiple ideas horizontally, then each of these ideas could be subjected to the second and third parts of the test without any great difficulty.

256. See *supra* text accompanying note 186; see also *Apple Computer, Inc. v. Microsoft Corp.*, 717 F. Supp. 1428, 1433 (N.D. Cal. 1989) (plaintiff submitted list of 189 similarities in particular features between copyrighted and allegedly infringing works).

257. The court applies the various limiting doctrines of copyright to ensure that only expression and not ideas are protected. See *supra* note 187 and accompanying text.

258. Even Judge Learned Hand himself realized that "[d]ecisions [will] inevitably be *ad hoc*." *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960). Nevertheless, the Abstractions Test is an attempt to assist courts in their determinations—and to make their decisions less arbitrary. See *supra* note 87–91 and accompanying text.

259. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2d Cir. 1992); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 215 (D. Mass. 1992) (*Borland II*) (discussing the *Paperback* test).

260. See *Altai*, 982 F.2d at 705 ("[E]ach subroutine is itself a program, and thus, may be said to have its own 'idea . . .'").

261. See *Borland II*, 799 F. Supp. at 216 (describing the multiple ideas of the 1-2-3 program in terms of specificity). Note that both *Paperback* and the *Borland* cases were decided by Judge Keeton, involved similar issues, and applied the same test.

262. The court has been criticized for not having recognized multiple ideas horizontally, though not in those terms. See Thomas K. Pratt, Comment, A Legal Test for the Copyrightability of a Computer Program's User Interface, 39 Kan. L. Rev. 1045, 1065 n.152 (1991) ("The [*Paperback*] court made the mistake of referring to the program's idea when it should have referred to the idea of the alleged copyrightable element of the user interface."). In fact, the court seems unable to avoid unwittingly recognizing multiple ideas horizontally. For example, it speaks of both the "idea for an electronic spreadsheet" as well as the "idea for a two-line moving cursor menu . . . used in a wide variety of computer programs including [but not limited to] spreadsheet programs." *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 65 (D. Mass. 1990).

The *Whelan* test, in contrast, is incapable of recognizing multiple ideas horizontally.²⁶³ The essence of the test is its simplicity: a program's purpose is its idea and whatever is not necessary to that idea is protected as expression.²⁶⁴ To the extent it is modified to allow for multiple ideas horizontally, it becomes an entirely new test. As for the *Brown Bag* test, it is not entirely clear whether it was intended to recognize multiple ideas at all.²⁶⁵ Because it does not even implicitly apply the Abstractions Test, the question never comes up—and is therefore never resolved. The weight of the evidence suggests that the test is not intended to recognize multiple ideas horizontally.²⁶⁶ Nevertheless, it would not fundamentally alter the test to have it do so. When, during the "extrinsic test," the court is analyzing the elements alleged to be similar, it could easily conclude that some of those elements are ideas themselves rather than elements of expression,²⁶⁷ since to try to fit such elements into the doctrines of merger or *scenes a faire* simply may not be appropriate in every case.²⁶⁸ Thus, only the *Whelan* test is incapable of recognizing multiple ideas horizontally.

263. It has already been argued that the *Whelan* test is able to accommodate multiple ideas vertically. See *supra* notes 251–255 and accompanying text.

264. See *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986).

265. The original "extrinsic test" was phrased as a "test for similarity of ideas." *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1164 (9th Cir. 1977). This wording is ambiguous: it is unclear whether it means (1) the similarity of the ideas in A as compared to those in B, or (2) the similarity of the idea of A to the idea of B (collectively, the "ideas"). As the "extrinsic test" became an "objective . . . analysis of expression," *Shaw v. Lindheim*, 919 F.2d 1353, 1357 (9th Cir. 1990), any pressing need for clarification was lost.

266. Despite the ambiguity referred to in the preceding footnote, the test seemed to assume that there was only one idea, for the court continually made reference to "the idea" of a work—in the singular. See *Sid & Marty Krofft*, 562 F.2d at 1163–65.

For a particularly striking example of the Extrinsic-Intrinsic Test being applied in a way that does not recognize multiple ideas horizontally, see *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. 616, 628–29 (N.D. Cal. 1993) (finding no substantial similarity as matter of law without looking to particular elements of expression because idea of copyrighted work "could not be more different" from idea of allegedly infringing work).

267. It is not entirely clear that this does not occur under the present formulation of the test. The district court in *Brown Bag*, for example, spoke of "the ideas [plural] inherent in [certain] features" of the program, and also of the idea of "pull down windows." *Telemarketing Resources v. Symantec Corp.*, 12 U.S.P.Q.2d (BNA) 1991, 1995 (N.D. Cal. 1989) (emphasis added). There is a fine line between idea and expression generally; it is all the more difficult to distinguish an idea from an expression that has merged with an idea.

268. A simple example would be the recent practice of including a spell-checking feature in spreadsheet programs. The average person would see this as a good idea, but if a program has only one idea horizontally, it becomes an expression. Moreover, it would be difficult to say that spell-checking "merges" with the idea of a spreadsheet program, or is in any way necessary for a spreadsheet program, since spreadsheets did rather well for a long time without them. By comparison, spell-checkers have long been standard on word processors and may well have "merged" with, or at least be *scenes a faire* to, the idea of a word processing program. However, without recognizing spell-checking as an idea in

Another line of comparison looks to the amount of protection afforded by each test. The *Whelan* and *Paperback* decisions have been criticized as providing too much protection.²⁶⁹ However, in the case of *Paperback*, the criticism is generally directed not at the *test itself*, but rather at the *result* reached by the court.²⁷⁰ Since the present concern is with the *tests themselves*, such criticism need not concern us at this point.²⁷¹ However, *Whelan* is a different matter. *Whelan* only allows for one idea in a computer program, and this is broadly defined as the program's purpose or function; everything that is not necessary to that idea is protected.²⁷² Very little can ever be said to be *necessary* to a broadly defined idea. Thus, a great deal of protection is inherently provided by the *Whelan* test. To the extent that the "intrinsic test" in *Brown Bag* is a "total concept and feel" test,²⁷³ it, too, has been criticized for being over-protective in that "total concept and feel" is a very broad concept that can easily blur the distinction between expression and idea.²⁷⁴ However, because the "intrinsic test" seems to be dependent upon the results obtained from the "extrinsic test," it is not clear that there is very much room for over-protection in the *Brown Bag* test.²⁷⁵

The *Altai* decision, on the other hand, has been criticized as being underprotective.²⁷⁶ A persuasive argument can be made on this point: because the "successive filtering method" does such a thorough job in filtering out unprotectable elements, very little remains for the comparison stage that could support a finding of substantial similarity. The "extrinsic test" of *Brown Bag* is susceptible to the same criticism: it, too, filters out unprotectable elements before comparison in the "intrinsic stage."²⁷⁷ Although the filtering is justifiable under the various limiting

itself, it could easily be seen as an original element of expression in a spreadsheet program. If so, it could be copyrighted and no other spreadsheet programs would be able to include a spell-checking feature.

269. See *supra* text accompanying notes 138-146 and *supra* text accompanying notes 169-176.

270. See *supra* text accompanying note 169.

271. The question of whether the 1-2-3 user interface should be copyrightable is really a policy question independent of whether the *Paperback* test is compatible with, say, the *Altai* test; that policy question is taken up later, see *infra* subsection III.D.2.

272. See *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986).

273. See *supra* notes 191-194 and accompanying text.

274. See *supra* note 202 and accompanying text.

275. It seems that the "intrinsic test" only considers those elements of expression that were not eliminated from protection by the "extrinsic test"—though this is not entirely clear. See *supra* notes 190, 192, 196-197 and accompanying text. If that is the case, then "total concept and feel" is not nearly as broad as it would be if it could consider the work as a whole, including unprotected elements.

276. See Raymond T. Nimmer, *supra* note 30, Special Update 2-5 ("The *Altai* test reduces protection for nonliteral elements of a program."); see also Recent Case, *supra* note 233, at 510-11 (arguing that the *Altai* test will "leave non-literal elements of computer programs under-protected").

277. Again, it is not entirely clear that this is the case. See *supra* note 275.

doctrines of copyright—especially because of the limited range of potential expression in computer programs²⁷⁸—the overall result of an approach such as the *Altai* test and possibly the *Brown Bag* test seems to be inherently underprotective.

Any comparison of the *Paperback* and *Altai* tests makes it readily apparent that the two tests are very similar.²⁷⁹ In the first step of each test, the court applies the Abstractions Test;²⁸⁰ in the second step, the court eliminates unprotectable elements from consideration;²⁸¹ and in the third step, the court evaluates the remaining elements to determine whether copyright infringement has occurred.²⁸² The major difference between the two tests lies with the second step. The second step of the *Paperback* test exempts only “elements essential to the expression of [a given] idea.”²⁸³ The *Altai* test breaks this step down to consider separately elements from the public domain, the doctrine of merger, and the doctrine of *scenes a faire*—and then further dissects the concept of *scenes a faire* in the context of computer programs.²⁸⁴ This represents a very sophisticated understanding of the limiting principles of copyright—but it also leads to underprotection.

The *Brown Bag* test is also fairly similar to the *Altai* test in that it too filters out any and all unprotectable elements.²⁸⁵ However, the *Altai* test has the advantage of organization: the court presented all the doctrines in one multi-step test, with specific guidance for the computer program context.²⁸⁶ *Brown Bag* never does this and therefore provides very little guidance for the courts that have to apply its tests.²⁸⁷ This, however, is not an inherent feature of the *Brown Bag* test; a subsequent court could easily provide greater organization and guidance.

Lastly, the *Altai* court, unlike the courts in *Paperback* and *Brown Bag*, limited the applicability of its test to a great, and uncertain, degree. The *Altai* test was adopted to determine the copyrightability of a program's

278. See supra notes 241–242 and accompanying text.

279. Indeed, Judge Keeton, who developed the *Paperback* test, stated that the two tests are “compatible substantively, though different in methodology.” *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 212 (D. Mass. 1992) (*Borland II*).

280. Compare supra notes 155–156 and accompanying text with supra notes 218–219 and accompanying text.

281. Compare supra notes 158–159 and accompanying text with supra notes 222–226 and accompanying text.

282. Compare supra note 163 and accompanying text with supra note 230 and accompanying text. For Judge Keeton's own comparison of the two tests, see *Borland II*, 799 F. Supp. at 211–12.

283. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 61 (D. Mass. 1990).

284. See supra notes 222–226 and accompanying text.

285. See supra notes 184, 187 and accompanying text.

286. See supra text accompanying notes 222–226; supra notes 240–244 and accompanying text.

287. See supra text accompanying notes 203–205.

structure in particular, and not that of any other nonliteral elements.²⁸⁸ The court expressed the following reservation: "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" ²⁸⁹ It is unclear how this will be interpreted, but it might be understood to exclude the user interface of a program from protection under this test.²⁹⁰ In this respect, the *Altai* test might be significantly different—i.e., less protective—than the *Paperback* and *Brown Bag* tests.²⁹¹

Statements made by the district court in the *Altai* case are even stronger: "Since the behavior aspect of a computer program [—the program as it functions in the computer, rather than the program as written—] falls within the statutory terms 'process', 'system', and 'method of operation', it may be excluded by [17 U.S.C. statute 102(b)] from copyright protection."²⁹² The user interface could be understood to be a "behavior aspect of a computer program," since it is the means by which the user communicates with the computer. Thus, it might actually be considered entirely uncopyrightable under *Altai*.²⁹³

However, there is nothing intrinsic to the *Altai* test that requires such reservation and limitation.²⁹⁴ For instance, the Nimmer treatise did not so qualify its own test, the "successive filtering method,"²⁹⁵ on which the

288. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 703 (2d Cir. 1992) ("We must determine the scope of the copyright protection that extends to a program's *non-literal structure* [only].") (emphasis added); see also Raymond T. Nimmer, *supra* note 30, Special Update 2 (The test was "adopted *solely* to measure infringement in program code and structure. The court took pains to emphasize that its discussion does not apply to other forms of copyright infringement . . .").

289. *Altai*, 982 F.2d at 703.

290. If, as will be argued later, the user interface is better understood as a structure than as an audiovisual work, excluding the user interface from protection under this test and consigning it to protection as an audiovisual work may effectively exclude it from protection entirely. See *infra* subsection III.D.2.

291. The *Paperback* test obviously provides protection to the user interface—that was the holding of the case. The Extrinsic-Intrinsic Test of the Ninth Circuit has also been held to protect the user interface of a computer program—in theory—if it is sufficiently expressive in a given case. See *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989). Whether this will prove to be the case *in fact* after *Brown Bag* remains to be seen.

292. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 560 (E.D.N.Y. 1991).

293. The Second Circuit's approving reference to this general section of the district court's opinion, *Altai*, 982 F.2d at 705–06, highlights the reality of such concerns.

Such a result would be at odds with *Paperback*'s decision to protect the user interface. See *supra* text accompanying note 162. In addition, it would likely not be the correct decision. See *infra* subsection III.D.2.

294. The court may have imposed such limitations, but even if it did so, the test would not be fundamentally altered if the limitations were subsequently dropped.

295. See generally 3 Nimmer on Copyright, *supra* note 16, § 13.03[F], at 13-102.9 to -102.35.

Altai test is based.²⁹⁶ It seems that the *Altai* court was merely being cautious because of the dynamic nature of the computer industry.²⁹⁷ Although this is responsible behavior on the part of the court—to decide only what is necessary and to limit its holding accordingly—there is no reason for the test to be bound by these reservations. In fact, the court recognized that under its newly announced test, “the exact contours of copyright protection for non-literal program structure [we]re not completely clear,” and “trust[ed] that as future cases are decided, those limits w[ould] become better defined.”²⁹⁸ The court even stated that, in an appropriate case, its decision “should not be read to foreclose the district courts . . . from utilizing a modified version [of the test].”²⁹⁹ Thus, the *Altai* test need not be interpreted as being so fundamentally limited as a first reading might suggest; the test can survive without the caution and rigidity of the opinions.

The foregoing analysis suggests that the four tests are not all compatible with each other. The *Whelan* test bears no real resemblance to any other test; it is a unique and simple test that cannot be reconciled with the others. The *Brown Bag* test is also significantly different from the others, though it is somewhat similar to the *Altai* test. The *Paperback* and *Altai* tests, however, are reconcilable: if the *Altai* decision is not interpreted strictly and the second step of the *Paperback* test is expanded, the tests become remarkably similar.

B. Which Approach Is Best?

The ideal test would be both simple to apply and accurate in its effect—i.e., neither overinclusive nor underinclusive. Often, however, some tradeoff must be made between simplicity and “accuracy.” The *Whelan* test has the advantage of simplicity;³⁰⁰ the court need merely determine the program’s function to arrive at its idea, and then determine what is unnecessary to that idea to arrive at the protected expression.³⁰¹ The other three tests are more complicated.³⁰² However, the desire for simplicity cannot overcome the need for accuracy.³⁰³ Especially in a field

296. See *supra* note 217.

297. See *Altai*, 982 F.2d at 706.

298. *Id.* at 712.

299. *Id.* at 706.

300. See Englund, *supra* note 145, at 881 (“Perhaps the single virtue of the *Whelan* rule is that it is easy to apply.”).

301. See *supra* text accompanying note 132.

302. Compare the *Whelan* test, *supra* text accompanying notes 131–133, with the *Paperback* test, *supra* text accompanying notes 154–165, the *Brown Bag* test, *supra* text accompanying notes 182–195, and the *Altai* test, *supra* text accompanying notes 217–231. The *Altai* test seems the most complicated.

303. Cf. *Basic, Inc. v. Levinson, Inc.*, 485 U.S. 224, 236 (1988) (“[E]ase of application alone is not an excuse for ignoring . . . Congress’ policy decisions.”).

The *Whelan* court itself agreed to this in principle: “Ease of application is not, however, a sufficient counterweight to the considerations we have adduced on behalf of

as complex as computer technology, it is foolish to believe that a simple resolution of delicate issues is likely to be adequate.

In fact, the *Paperback* and *Altai* tests are not unduly complicated. By breaking down an enormous task—that of deciding which nonliteral elements of a computer program should be protected—into smaller, pre-defined subtasks—the application of particular copyright doctrines such as merger or *scènes a faire*—problem solving becomes a much simpler process.³⁰⁴ Although the *Brown Bag* test is similar in many respects, it never adequately defines the steps involved.³⁰⁵ The result is a test that is difficult to apply—which promotes neither simplicity nor accuracy.³⁰⁶ Significantly greater guidance is needed from the Ninth Circuit to overcome this problem. By contrast, the *Whelan* test is wholly inadequate in terms of accuracy: it is hopelessly overbroad.³⁰⁷ Thus, an approach similar to the *Paperback* or *Altai* tests is superior to the others in terms of the trade-off between simplicity and accuracy.

A second consideration in evaluating these tests is the extent to which each test furthers the “promot[ion of] the Progress of Science and useful Arts.”³⁰⁸ Optimally, the law should strike a balance between the incentive to create that comes from the prospect of personal gain³⁰⁹ on the one hand, and the ability to improve on the ideas of predecessors on the other hand. CONTU³¹⁰ believed that this should hold true for computer programs. However, it insisted that “[c]opyright should not grant anyone more economic power than is necessary to achieve the incentive to create.”³¹¹ Since this is fundamental to copyright law, every court making a copyright decision must believe that its approach furthers this goal—and the courts have stated as much.³¹²

our position.” *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1238 (3d Cir. 1986).

304. Any computer programmer would attest to this fact. Compare the process of developing a complex computer program, described at *supra* text accompanying notes 20–28.

305. See *supra* notes 203–205 and accompanying text.

306. In addition, the district courts will each be forced independently to interpret the *test itself*, which will lead to inconsistent results.

307. See *supra* text accompanying notes 138–143, 263–264.

308. U.S. Const. art. I, § 8, cl. 8.

309. See *Mazer v. Stein*, 347 U.S. 201, 219 (1954) (“[E]ncouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors.”).

310. See *supra* text accompanying notes 49–50.

311. CONTU Final Report, *supra* note 50, at 12.

312. See *Computer Assocs. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693, 711 (2d Cir. 1992) (“We are satisfied that the three step approach we have just outlined not only comports with, but advances the constitutional policies underlying the Copyright Act.”); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1237 (3d Cir. 1986) (“The rule proposed here . . . 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.”); *Lotus Dev. Corp. v. Paperback*

The question of which approach best advances the goal of copyright law is difficult to answer. It may be that computer programs are in need of a great deal of protection because of the ease with which they may be copied or imitated; yet it is equally possible that existing financial incentives alone are sufficient to stimulate innovation and therefore eradicate the need for protection.³¹³ This question is simply not capable of theoretical resolution. Therefore, the courts are in need of legislative guidance to make the necessary policy decision. In the meantime, however, the courts *are* forced to make difficult decisions.³¹⁴ Yet there is no consensus as to what amount of protection efficiently promotes progress in the computer software industry. The four decisions demonstrate that divergent opinions as to what is appropriate lead to different approaches. While opinions abound as to which test is most likely to promote progress, none of the approaches has been empirically proven to be misguided.

In the absence of any indication that Congress intended special treatment for computer programs,³¹⁵ the courts should look to well-established principles of copyright in order to determine which aspects of a computer program should be considered copyrightable. These principles have guided the courts well throughout the years in the promotion of progress in the face of technological developments and should continue to do so in this dynamic area.

Overall, the *Altai* test displays the richest understanding of the various concepts that define and limit copyrightability, particularly in the computer program context. The *Brown Bag* test has the advantage of being a general test for copyrightability:³¹⁶ it was not specifically created for the computer context. However, it is a difficult and confusing test.³¹⁷ In addition, it does not adequately assess the interrelationship between idea and expression in a copyrighted work.³¹⁸ The net result is a test that could hardly be applied in a consistent manner. The *Whelan* test is entirely too simplistic and amounts to only a rough amalgamation of the

Software Int'l, 740 F. Supp. 37, 78 (D. Mass. 1990) ("The encouragement of innovation requires no more.").

313. Cf. *Paperback*, 740 F. Supp. at 46 (discussing alternate possible paths Congress could take to further constitutional goal).

314. Cf. *id.* at 53 ("Rather than itself drawing the boundary line between copyrightable and non-copyrightable elements of computer programs, Congress has mandated that courts . . . determin[e] this boundary line.").

315. The CONTU Final Report suggested no special treatment for computer programs. See *supra* note 62 and accompanying text.

316. This is not entirely true. The Extrinsic-Intrinsic Test as modified by *Shaw v. Lindheim*, 919 F.2d 1353 (9th Cir. 1990)—i.e., the *Brown Bag* test—"applies by its own terms only to literary works." *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1476 (9th Cir. 1992).

317. See *supra* notes 305–306 and accompanying text.

318. This is because it does not apply the Abstractions Test. See *supra* notes 256–258 and accompanying text.

concepts of merger and *scenes a faire*.³¹⁹ The *Paperback* test reflects a more precise understanding of these concepts³²⁰ and applies the Abstractions Test as well.³²¹ However, only the *Altai* test *explicitly* incorporates in a *systematic* way the Abstractions Test, the concept of merger, a very sophisticated understanding of the concept of *scenes a faire*,³²² and the concept of public domain materials.³²³ In short, from the perspective of copyright doctrine, the *Altai* test is superior to the others.

However, the *Altai* court's severe limitation of the applicability of its test³²⁴ detracts from its doctrinal purity: these limits are based on considerations other than copyright principles.³²⁵ Fortunately, this aspect of the *Altai* opinion is not intrinsic to the test that it announced;³²⁶ thus, it could be eliminated without substantially altering the nature of the test.³²⁷ Moreover, despite *Altai*'s elaboration on how copyright doctrines should apply in the computer program context, there is no reason why its test could not be used as a general test for substantial similarity; in fact, Professor Nimmer believes that his "successive filtering method"—on which the *Altai* test is based—should be.³²⁸ Thus, the *Altai* test can provide a framework based entirely on existing copyright principles.

In conclusion, if the determination as to which test is the "best" is to be made on broad policy grounds, then the conclusion will be based simply on each decision-maker's subjective belief as to the likely effects of each test. There is no means by which such effects can be deduced logically; even empirical observation would likely not lead to obvious conclusions. However, if the determination is to be based on copyright doctrine, a principled decision is possible. The *Altai* test—if not the *Altai* opinion—presents the best framework for an analysis of substantial similarity. The only remaining problem is the underprotection *inherent* to the test, a problem discussed in the next section.

319. See *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1235–36 (2d Cir. 1992) (discussing the *Baker* case and *scenes a faire*).

320. The second stage of this test explicitly allows not only for "elements essential to expression of [an] idea," but also for elements which are "one of only a few ways of expressing the idea." *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 61 (D.Mass. 1990). Compare *id.* with *Whelan*, 797 F.2d at 1236.

321. For a review of the Abstractions Test, see *supra* text accompanying notes 87–91.

322. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 709–10 (2d Cir. 1992); see also 3 Nimmer on Copyright, *supra* note 16, § 13.03[F][3], at 13-102.21 to -102.28. The external constraints considered in this sophisticated understanding of *scenes a faire* are briefly discussed at *supra* notes 240–244 and accompanying text.

323. See *supra* text accompanying notes 222, 226.

324. See *supra* notes 288–292 and accompanying text.

325. See *supra* text accompanying note 297.

326. See *supra* text accompanying notes 294–299.

327. As mentioned in the previous section, to the extent that those limitations are eliminated, the *Altai* test begins to take on the character of the *Paperback* decision. See *supra* section III.A.

328. See 3 Nimmer on Copyright, *supra* note 16, § 13.03[F], at 13-102.9 to -102.35.

C. Reincorporating Compilations

An inescapable failing of the *Altai* test is that its thoroughness in filtering out unprotectable elements of expression leads to underprotection. Due to the rigors of the underlying test, perhaps even a tempered approach by the court could not escape its strictness. For example, suppose that a programmer develops a program that is composed entirely of elements that are uncopyrightable: part of the program was dictated by efficiency concerns, part by external factors to some degree or another, and part comes from the public domain. Assume further, however, that the programmer has selected, coordinated, and arranged these uncopyrightable elements in the program in a creative and original manner. Under even a tempered *Altai* approach, this program would be afforded absolutely no protection by copyright. This result is inappropriate.

It might seem at first that the *Brown Bag* test solves this problem through the "intrinsic test." It does not, however, since summary judgment may be granted by the court if there are no objective similarities in the individual elements of expression.³²⁹ Thus, the intrinsic test would never be triggered in the above example. Moreover, even when the "intrinsic test" is triggered, it is not clear that the work as a whole may be considered.³³⁰ If only the protected elements may be considered in the "intrinsic test," then *Brown Bag* is no better in this respect than *Altai*.

Application of the compilations doctrine³³¹ rectifies this failing. As mentioned above, works consisting entirely of uncopyrightable elements are often protected by the copyright law under the rubric of "compilations."³³² Based on an analogy to compilations,³³³ copyright protection should subsist in computer programs even if they consist entirely of uncopyrightable elements if those elements were selected or arranged in an original manner. Therefore, an additional step should be added to *Altai*'s "Abstraction-Filtration-Comparison" test. After the court has completed the successive filtering of the second step of the *Altai* test, it should re-evaluate the elements that have been excluded as uncopyrightable to determine whether there are elements, or groupings of elements, that, in

329. See *supra* note 190 and accompanying text.

330. See *supra* note 192.

331. See *supra* text accompanying notes 92-98.

332. See *supra* text accompanying notes 92-98; see also *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 111 S. Ct. 1282, 1287-89 (1991) ("[E]ven a directory that contains absolutely no protectible written expression, only facts, meets the constitutional minimum for copyright protection if it features an original selection or arrangement").

333. This would not be unprecedented. See, e.g., *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990):

If particular characteristics not distinctive individually have been brought together in a way that makes the "whole" a distinctive expression of an idea . . . then the "whole" may be copyrightable. The statutory provisions regarding "compilation," . . . are not essential to this conclusion, but do reinforce it.

Id. at 67; see also *infra* text accompanying notes 340-342, discussing "total concept and feel."

their selection, coordination, or arrangement, are sufficiently original to merit copyright protection as a group in a manner analogous to compilations. This would be a "reincorporation" stage since elements previously excluded from copyright protection are now reincorporated in the analysis, albeit as compilations.

The leading Supreme Court decision on the subject of compilations is *Feist Publications, Inc. v. Rural Telephone Service Co.*³³⁴ There, the court stated that, while originality is a requirement for copyright protection, it requires "only that the author make the selection or arrangement [of the pre-existing materials] independently (i.e., without copying that selection or arrangement from another work), and that it display some minimal level of creativity."³³⁵ The level of creativity required is low; "[p]resumably, the vast majority of compilations will pass this test . . ."³³⁶

The *Feist* opinion provides guidance for the reincorporation stage. The courts should take notice of the fact that programs, even if consisting entirely of uncopyrightable elements, can be "original works of authorship" deserving of copyright protection if the selection, coordination, or arrangement of those elements in the program is original. Although most programs would receive protection under this approach, this result need not be feared as overly protective. The *Feist* court stated that compilations receive only limited protection from copyright law.³³⁷ Thus, the protection that a computer program would receive would also be limited: only the elements that owe their origin to the programmer would be protected³³⁸—i.e., the selection, coordination, and arrangement of the uncopyrightable elements in the program, but not the uncopyrightable elements themselves. Any elements that are afforded protection *only* as a compilation would thus receive significantly less protection than they would otherwise: Copyright infringement of compilations "consisting largely of uncopyrightable elements . . . should not be found in the absence of 'bodily appropriation of expression.'"³³⁹

This proposal bears resemblance to the "total concept and feel" doctrine,³⁴⁰ but the compilations approach is superior. The concern behind "total concept and feel" is the same: that even if the individual elements comprising a work are not protectable, still the work as a whole may de-

334. 111 S. Ct. 1282 (1991). Although this case deals specifically with compilations of facts, as opposed to compilations of other "materials," the copyright law makes no distinction between compilations of data (i.e., facts) and compilations of other materials. See 17 U.S.C. § 102(a) (1988). Thus, the *Feist* opinion should not be interpreted as being limited to the context of factual compilations but should be considered applicable to compilations generally.

335. *Feist*, 111 S. Ct. at 1294–95.

336. *Id.* at 1294.

337. See *id.* at 1295.

338. See *id.*; see also *supra* text accompanying notes 96–98.

339. *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. 616, 623 (N.D. Cal. 1993) (quoting *Harper House, Inc. v. Thomas Nelson, Inc.*, 889 F.2d 197, 205 (9th Cir. 1989)).

340. See *supra* notes 99–104 and accompanying text.

serve protection.³⁴¹ The problem with the "total concept and feel" doctrine is that it is amorphous—it is a non-test that allows gut reaction to prevail over critical analysis. In doing so, it comes dangerously close to protecting ideas.³⁴² The addition of the compilations doctrine expands the scope of the *Altai* test and accommodates these concerns while simultaneously avoiding the overbreadth of "total concept and feel." It strikes the appropriate balance between the two, providing the substance needed for analysis while ensuring that only expression is protected.

One final point should be made with regard to the proposed reincorporation stage. In applying it, the court should recognize the fact that a work may be held to consist of *more than one* compilation, just as a work "may consist of a mixture of numerous ideas and expressions."³⁴³ For example, the overall structure of the program may be one compilation,³⁴⁴ the user interface may be another, and a database included with the program—i.e., the standard type of compilation—may be yet a third. Each "compilation" would simply be one more element to be evaluated in the final comparison stage.

With the elimination of the unnecessary limitations imposed by the court³⁴⁵ and the addition of this reincorporation stage, the *Altai* test becomes complete. In theory, at least, it should provide protection to *all* the elements and *only* to the elements of a computer program that deserve protection. Although decisions will still be "ad hoc" to a certain extent, this Modified *Altai* Test provides the decision-maker with a framework for determining copyrightability that is solidly grounded in well-established copyright principles.

D. *Re-Evaluation of the Concerns Often Raised Regarding Copyrightability*

There are two ways that two works may be substantially similar without being identical. Most people are aware that "fragmented literal similarity"³⁴⁶—limited but literal copying—is prohibited; but "it is equally clear that two works may not be literally identical and yet, for the purposes of copyright infringement, may be found to be substantially similar."³⁴⁷ This is the case of "comprehensive nonliteral similarity": one work duplicates the essence or structure of another without any literal

341. See *Roth Greeting Cards v. United Card Co.*, 429 F.2d 1106, 1109–11 (9th Cir. 1970).

342. See *supra* notes 104, 200–202.

343. See 3 Nimmer on Copyright, *supra* note 16, § 13.03[F][1], at 13-102.18.

344. When the structure of a program is protected only as a compilation, it will receive limited protection against only detailed copying. See *supra* text accompanying note 339.

345. See *supra* text accompanying notes 294–299.

346. See generally 3 Nimmer on Copyright, *supra* note 16, § 13.03[A][2], at 13-46 to -54.

347. *Id.* § 13.03[A], at 13-27 to -28.

copying.³⁴⁸ The courts have wisely noted that such copying is prohibited generally and should therefore also be prohibited with regard to computer programs.³⁴⁹

1. *Whether the Structure of a Computer Program Should Be Protected.* — Since there may be “comprehensive nonliteral similarities” between the structures of two programs, there should be no question but that the structure of computer programs should be protected—if in any given case it is sufficiently original.

Many commentators have expressed fears that the *Whelan* decision went too far in extending protection to the structure of computer programs.³⁵⁰ However, if *Whelan* is interpreted to provide protection only to detailed structures³⁵¹—and only when they are not necessary to the program³⁵²—then these concerns should be allayed to a great extent.

Regardless of whether the *Whelan* test would provide too much protection, however, the Modified *Altai* Test certainly does not. The structure of a program would be protected only if it withstood the successive filtering of uncopyrightable elements: elements dictated by efficiency, or by external factors, or elements taken from the public domain would not be protected. But, if there remains after all of this elimination “a core of protectable expression,”³⁵³ or if despite the lack of such a core, the selection, coordination, or arrangement is sufficiently original to deserve protection,³⁵⁴ then there would be no reason to exclude the structure from copyright protection.

It certainly may be the case that from a policy perspective even this is too much protection. However, the likely effect of the successive filtering of the Modified *Altai* Test is that few structures will be protected—and even then only against the most detailed of copying. This is not likely to be overprotective. On the other hand, this may provide too little protection, but this also seems unlikely. Prudence dictates that because of the limited range of potential expression inherent to computer program-

348. See *id.* § 13.03[A][1], at 13–29 (“By this is meant a similarity not just as to a particular line or paragraph or other minor segment, but where the fundamental essence or structure of one work is duplicated in another.”). An example of “comprehensive nonliteral similarity” is a similarity in the plot or the major characters of two stories.

349. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 701 (2d Cir. 1992); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1234 (3d Cir. 1986); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 51–52 (D. Mass. 1990). While the *Brown Bag* court did not do so explicitly, this is clearly implicit throughout the opinion. See *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465 *passim* (9th Cir. 1992).

350. See *supra* notes 144–146 and accompanying text.

351. The *Whelan* court stated that “the detailed structure of the Dentalab program is part of the expression . . . of that program.” 797 F.2d at 1239; see also Raymond T. Nimmer, *supra* note 30, Special Update 1–3 (“The court’s review hinged . . . on the particularity and detail of copying.”).

352. See *Whelan*, 797 F.2d at 1236.

353. *Altai*, 982 F.2d at 710.

354. See *supra* section III.D.

ming,³⁵⁵ the protection afforded to the structure of computer programs *should* be less than that afforded to the structures of other literary works with greater potential for creativity. That result is also dictated by the doctrines of merger and *scenes a faire*.³⁵⁶

Regardless of the resolution of that issue, the structure of literary works is protected under copyright law;³⁵⁷ thus, the structure of a computer program must also be protected. However, since computer programs were not afforded special protection under the 1976 Act,³⁵⁸ their structures should be protected only to the extent they survive scrutiny under the Modified *Altai* Test.

2. *Whether the User Interface Should Be Protected.* — A user interface relies heavily on screen displays.³⁵⁹ Therefore, it might be argued that the user interface should be protected, if at all, under the category of “audiovisual works” rather than “literary works.”³⁶⁰ However, although the screen displays of a program may also be copyrightable as an audiovisual work, the user interface consists of much more than mere screen displays.³⁶¹ It is possible for two programs to have user interfaces that function identically and yet have very different screen displays.³⁶² Thus, a copyright for screen displays would not provide sufficient protection for the user interface; it is often the *substance* of the user interface—e.g., the hierarchical organization of the function menus—rather than merely the

355. See *supra* notes 241–242 and accompanying text.

356. In addition, the result would also seem to be dictated by the requirement of originality, see *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 111 S. Ct. 1282, 1288 (1991) (“Originality is a constitutional requirement.”), since, if there is a limited range of potential expression, the threshold for originality must be raised.

357. See *supra* text accompanying notes 346–349.

358. In passing the 1976 Act, Congress decided to wait for the CONTU recommendations before taking any action on the matter. See *supra* text accompanying notes 49–53. The CONTU Final Report did not suggest any special treatment for computer programs. See *supra* text accompanying notes 62–64.

359. See *supra* text accompanying note 30.

360. Support for this proposition can be found in *Stern Elecs., Inc. v. Kaufman*, 669 F.2d 852 (2d Cir. 1982). In *Stern*, the defendant had manufactured a “knock-off” version of plaintiff’s “Scramble” video game without actually copying any of the code. The court rejected defendant’s assertion that “only the written computer program” was entitled to protection. *Id.* at 855. Thus, it extended protection to the nonliteral elements of the program—but it did so using the category of “audiovisual work” rather than “literary work.” The court also held that “the player’s participation does not withdraw the audiovisual work from copyright eligibility.” *Id.* at 856. Based on this case it could be argued that the output of an interactive user interface—which consists mostly of screen displays—should be protected as an audiovisual work rather than a literary work.

361. See *supra* notes 29–30 and accompanying text.

362. While the screen displays of the VP-Planner program in the *Paperback* case may have been virtually identical, the displays in the Quattro Pro program at issue in *Borland II* were very different. By converting the 1-2-3 user interface from a “two-line moving-cursor menu” system to a “pull-down bar menu[]” system, Borland achieved an entirely different look for its Quattro Pro program. Nevertheless, the Quattro Pro program copied the user interface’s substance in its entirety. This was rightly held to infringe Lotus’ copyright. See *Lotus Dev. Corp. v. Borland Int’l, Inc.*, 799 F. Supp. 203, 217–20 (D. Mass. 1992).

screen displays that are generated by it, that developers want protected.³⁶³ That substance is derived from the overall structure of a program—or from certain of its subroutines—and therefore deserves protection as a literary work.³⁶⁴

As the *Paperback* court demonstrated, a user interface may be expressive.³⁶⁵ Nevertheless, there is the possibility that a court, in the course of successively filtering out a program's uncopyrightable elements, may improperly decide that a particular user interface is not protectable.³⁶⁶ The Reincorporation stage proposed above³⁶⁷ allows the judge to re-evaluate such a decision given that the particular choices made in designing the user interface involve creative expression and originality and as such should be protected.

It should be noted that the current trends in the computer industry suggest that the compatibility/standardization concerns that are often raised³⁶⁸ are not likely to be as problematic in the future. Because of the surge in popularity of operating systems utilizing graphical user interfaces ("GUI")—beginning with the Macintosh, and now including OS/2 and Windows for the IBM PC—application programs are beginning to "look and feel" much more similar.³⁶⁹ Thus, users will be less "locked into" a specific program merely because the training costs involved in switching to a new system are prohibitive.³⁷⁰ Incidentally, this very development puts the nails on the coffin of "total concept and feel":³⁷¹ it becomes

363. At times, developers may want the screen displays protected. See the *Apple* cases, *supra* note 9. In such cases, the copyright should be dealt with as an audiovisual work. However, this should not obscure the fact that at times, the substance of the user interface, rather than the screen displays, is important. See the *Borland* cases, *supra* note 154. In such cases, the user interface ought to be treated as a nonliteral element of a literary work, the program. Although the test may be the same in either case, treating a user interface as an audiovisual work may improperly result in a lack of protection.

364. Congress intended computer programs to be protected as a literary work. See *supra* note 48 and accompanying text.

365. See *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 65–68 (D. Mass. 1990).

366. This could easily happen under a stringent application of any of the "filtering" concepts—merger, *scenes a faire*, or public domain.

367. See *supra* section III.C.

368. See *supra* notes 173–174 and accompanying text.

369. IBM developed SAA (Systems Application Architecture) standards to advance compatibility among computers and programs. See 3 Nimmer on Copyright, *supra* note 16, § 13.03[F][3][c], at 13-102.25. This standard has been incorporated into the Windows operating system:

Conformance to SAA standards has been most beneficial to users. New applications must now adhere to the Windows standard, thereby adhering to the consistent SAA standard. It has become easier and easier to quickly learn and use new applications. Many Windows applications have many of the same menu choices, and the methods of accessing and navigating the menus are the same.

Judd Robins, *Supercharging Windows* 12–13 (1992).

370. See, e.g., Verdesca, *supra* note 169, at 1076 ("Without compatibility, today's computer user becomes locked into a specific vendor's application.").

371. See *supra* notes 99–104 and accompanying text.

meaningless in the computer context when operating systems are *designed* to make applications "look and feel" similar. The similarity among application programs also has another effect: much of a user interface that operates under a GUI will be standardized³⁷² and to that extent will not be protected. Nevertheless, an application's user interface as a whole, which likely involves *some* originality in the selection, coordination, and arrangement of its features, would be copyrightable and protected *at least* from duplication of the *entire* work.³⁷³

To the extent, then, that a user interface is expressive and/or creative *as a whole*, it should be protected under copyright principles. Copying a user interface in its entirety is almost certainly an infringement of the original program's copyright because it would necessarily involve copying whatever original and hence copyrightable elements were present.

CONCLUSION

Of the four tests for determining the copyrightability of the nonliteral elements of computer programs provided in the *Whelan*, *Paperback*, *Brown Bag*, and *Altai* cases, a modified version of the *Altai* test is the best approach from the perspective of copyright principles. However, the question of which test would be the best from a policy perspective is difficult and not one in which the courts should heavily involve themselves.

Since the goal of copyright law is to "promote progress," however, a resolution of such policy issues is necessary. The only body capable of making a legitimate determination of those issues is Congress. In having established CONTU to assess those issues in the seventies, Congress was acting responsibly; in not having acted since—as, for example, by establishing a CONTU II—Congress has shown lamentable apathy. As these four cases show, the result has been the development of various incompatible standards. Congress must act soon to put an end to the uncertainty that prevails in this area. Congressional action on such issues would not only be welcome,³⁷⁴ but also well overdue.³⁷⁵

372. For instance, most Windows applications have a menu bar that begins with the command "File" which usually includes in its submenu, among other choices, "New," "Open . . .," "Close," "Save," "Save As . . .," "Print . . .," and "Exit"; the second command is usually "Edit," and includes in its submenu the choices "Undo," "Cut," "Copy," and "Paste"; et cetera. See, e.g., Microsoft Windows Write Version 3.1, WordPerfect for Windows Version 5.2, Quattro Pro for Windows Version 5.0, and Lotus 1-2-3 Release 4 for Windows.

373. See *supra* section III.C; cf. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 67 (D. Mass. 1990) ("The fact that some of these specific command terms are quite obvious or merge with the idea of such a particular command term does not preclude copyrightability for the command structure taken as a whole.").

374. See *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992) ("[N]ow that more than 12 years have passed since CONTU issued its final report, the resolution of this specific issue could benefit from further legislative investigation—perhaps a CONTU II.").

375. The CONTU Final Report, *supra* note 50, at 2, recommended that "[a]ny legislation enacted as a result of these recommendations . . . be subject to a periodic review to determine its adequacy in the light of continuing technological change."

In the meantime, however, the courts must struggle with this problem. The *Altai* test is a commendable effort at developing a test based on well-established copyright principles, but falls short in a number of important respects. The Modified *Altai* Test presented in this Note remedies these shortcomings and therefore provides the appropriate legal test to determine copyrightability in the absence of any legislative intent to provide special treatment for computer programs.

Julian Velasco