
Michael J. Bendel

Follow this and additional works at: http://repository.jmls.edu/jitpl

Part of the Computer Law Commons, Intellectual Property Law Commons, Internet Law Commons, Privacy Law Commons, and the Science and Technology Law Commons

Recommended Citation

http://repository.jmls.edu/jitpl/vol12/iss3/6

This Comments is brought to you for free and open access by The John Marshall Institutional Repository. It has been accepted for inclusion in The John Marshall Journal of Information Technology & Privacy Law by an authorized administrator of The John Marshall Institutional Repository.
AN ANALYSIS THAT IS NOT "AD HOC":
THE BIFURCATED UNIFORM ANALYSIS THAT THE FEDERAL COURTS SHOULD FOLLOW TO DETERMINE COMPUTER PROGRAM COPYRIGHT NONLITERAL INFRINGEMENT

I. INTRODUCTION

In the last ten years,1 the federal courts2 have started to determine the scope of protection the Copyright Act3 grants to the components in a computer program other than the text in which a computer program4

1. Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), is the leading case recognizing that the literal components of the computer program are entitled to copyright protection. Peter S. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 Stan. L. Rev. 1045, 1073 (1989). Apple was also the first to confront and reject the claim that the nonliteral components are entitled to copyright protection. 1 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 2.04(C) (1992). Just ten years later however, the majority of the federal courts now recognize that the nonliteral components are entitled to copyright protection. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 702 (2d Cir. 1992).

2. Because the Copyright Act, 17 U.S.C. §§ 101-914 (1988), is Federal law, the doctrine of preemption prohibits the state courts from determining the copyright infringement issues. Gates Rubber Co. v. Bando Am., Inc., 798 F. Supp. 1499, 1521 (D. Colo. 1992). Therefore, the federal courts, e.g., the district courts and the circuit courts of appeal, develop all the case law that arises under the Copyright Act.


The Copyright Act extends copyright protection to any subject matter that is listed in the Copyright Act, e.g., literary works, musical works, dramatic works, derivative works, etc. 17 U.S.C. §§ 102-05 (1988). Subject matter protected by the Copyright Act grants the owner of that copyrighted work the exclusive rights to reproduce copies, to prepare derivative works, to distribute copies, to perform the work publicly, and to display the work publicly. 17 U.S.C. § 106 (1988). In addition, the Copyright Act grants and limits other specific rights that depend on the nature of the work, e.g., rights to attribution and integrity, fair use, licensing, etc. 17 U.S.C. §§ 107-20 (1988).

Finally, the value of these rights is that they are a bundle of rights that can be sold one at a time to different buyers and the courts will protect any one or all of the rights from someone infringing their scope. See infra note 9 for a further discussion of the value of copyrights.

4. A "computer program" is defined in the Copyright Act (1980) as "a set of state-
is written. The components in a computer program can be classified into two groups. The first group is referred to as the *literal* textual component group. The textual source and object code of a computer program define the literal components of the program. The second group is referred to as the *nonliteral* component group. The nonliteral components include every component of the program that is not a literal component, for example, the overall organization of a program, the structure of a program's command system, and the sequence of the presentation of information on the computer screen. Today, the majority of the federal courts recognize that the nonliteral components of a computer program are entitled to copyright protection. Consequently, now the courts must define the scope of that protection. However, defining the proper scope of copyright protection is proving to be a bigger challenge than at least one court could have ever anticipated.

A "computer program" is commonly referred to as "software", and these two terms are used synonymously in this comment. Pearl Systems, Inc. v. Competition Elec., Inc., 8 U.S.P.Q.2d (BNA) 1520, 1520 n.1 (S.D. Fla. 1988).

Judge Learned Hand's idea that nonliteral components exist in a work has been adopted in the most recent computer program infringement case as follows: "[C]opyright protection extends beyond a literary work's strictly textual form to its non-literal components. . . . [I]t is of course essential to any protection of literary property that the right cannot be limited literally to the text, else a plagiarist would escape by immaterial variations." Computer Assocs., 982 F. 2d at 702 (quoting Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d. Cir. 1930)).

This result is stated clearly in the leading case on computer program nonliteral infringement: "If the nonliteral structures of literary works are protected by copyright; and if computer programs are literary works, as we are told by the literature; then the nonliteral structures of computer programs are protected by copyright." Computer Assocs., 982 F. 2d at 702.

**Note**

- 3. See also infra note 19 for a further discussion of the nature of a computer program.
- 4. A computer programmer writer writes the textual form of the program as a set of statements or instructions that a person can read line by line and figure out the possible results that a program will achieve when the program is run on a computer. See Randall Davis, *The Nature of Software and Its Consequences for Establishing and Evaluating Similarity*, 5 SOFTWARE L.J. 299 (1992).
- 6. The idea that nonliteral components exist in a copyrightable work was first created by Judge Learned Hand in *Nichols v. Universal Pictures Corp.*, 45 F.2d 119 (2d. Cir. 1930) (a literary works infringement case where the court held that a theater play is copyrightable beyond the mere text of the work).
- 7. Whelan was the first case to hold that the nonliteral and the literal components of a computer program are protectable under the Copyright Act. Whelan, 797 F.2d at 1248. However, the analysis that the court developed to reach its holding was flawed, be-
The difficulty of the scope challenge is evidenced by the federal courts' failure to adopt a uniform analysis for determining when the nonliteral components of one program are infringed by another program. The federal courts' failure presents two major problems. First, the grant of copyright protection is only valuable if that grant includes the right of enforceability in a court of law. Second, without a uniform analysis for the federal courts to follow, both computer technology and copyright law will suffer, because: (1) computer technology growth will be stunted since a computer program owner will not know (a) when a program may be infringed, or (b) when a newly created program may infringe another existing program; and, (2) copyright law will stagnate because the various federal circuits will define the scope of copyright protection differently.

Thus, in order to create an enforceable copyright “right” and to avoid the problems of non-uniformity created by “determining” the proper scope of copyright protection, the federal courts must adopt a uniform analysis to apply in computer program nonliteral infringement. See infra notes 99-104 and the accompanying text for a criticism of the Whelan analysis.

9. In copyright law, the right of enforceability of one's copyrights arises when "anyone violates any of the exclusive rights of the copyright owner as provided by sections 106 through 118. . ." 17 U.S.C. § 501 (1988). Violation of a user's permitted use, as defined in the Copyright Act, is termed an infringement of the copyrighted work. Id.

Section 117 of the Copyright Act defines the rights of the computer program copyright owner. 17 U.S.C. § 117 (1988). Together, section 117 and the rights granted in section 106, require the federal courts to adopt a uniform analysis for determining copyright nonliteral infringement, in order to grant the proper scope of protection to the nonliteral components in a computer program. See also infra notes 10-13 and the accompanying text discussing why a uniform analysis is necessary.

10. Computer technology grows one building block at a time. Davis, supra note 5, at 304-06. A new program cannot be expected to recreate ideas that already exist in the public domain, or else growth would occur at a rate too small to sustain the computer software industry. Whelan, 797 F.2d at 1238. On the other hand, a new program cannot merely copy the innovative expression that makes an existing program so valuable. Thomas M. Gage, Whelan Associates v. Jaslow Dental Laboratories: Copyright Protection for Computer Software Structure—What's the Purpose?, 1987 Wis. L. Rev. 859, 873 (1987).

11. The law does not advance very rapidly when there are several different rules that can be applied to similar fact patterns. Gates, 798 F. Supp. at 1510-11. See also infra notes 76-82 and the accompanying text for a discussion of the different tests the courts have already created to approach the computer program copyright nonliteral infringement issue.

Furthermore, the origin of copyright law in the Constitution suggests that the rules emanating from that single beginning should not contradict one another. U.S. Const. art. I, § 8, cl. 8.

12. The “determination” of copyright infringement is never achieved by simply finding the bright-line Congress drew in the Copyright Act, because Congress drew no such line. Lotus, 740 F. Supp. at 73. “Congress could have constitutionally drawn defendants' proposed bright-line—providing protection only to computer program code and not to nonliteral elements of computer programs—but Congress has not done so... Instead, the court must adjudicate within the area of protection mandated by Congress.” Id.
Throughout the development of nonliteral infringement case law, the courts have agreed on how to resolve the initial issues in a nonliteral infringement analysis by applying the traditional literal infringement rules. However, when a court must address the issues comprising nonliteral component copyrightability and nonliteral component substantial similarity, the courts' analyses are “ad hoc.”

Various courts have taken different approaches to resolve the copyrightability and the substantial similarity issues. However, all of

---

13. See infra notes 128-140 and the accompanying text discussing the proposed uniform analysis the courts should adopt.

14. In a copyright infringement suit, the plaintiff must prove (I) copyright ownership and (II) copying. 3 NIMMER, supra note 1, §§ 13.01-03. Copyright ownership requires the plaintiff to prove she owned a valid copyright to the program. E.F. Johnson, 823 F. Supp. 1485, 1491-92 (D. Minn. 1985). Copying requires the plaintiff to prove that the defendant copied the plaintiff’s program in creating his “own” program. Computer Assocs., 775 F. Supp. at 557-58. The plaintiff can prove copying through (A) direct evidence, e.g. the defendant outright admits having copied the plaintiff’s program, or through (B) indirect evidence. Id. Indirect evidence of copying requires the plaintiff to prove both (1) the defendant had access to a copy of the plaintiff’s program, and (2) the defendant illicitly copied the plaintiff’s program, that is, the quality and the quantity of the defendants copying violate the scope of protection the Copyright Act grants to the plaintiff’s program. Whelan, 797 F.2d at 1231-32.

In proving access and illicit copying, direct evidence is near impossible to obtain. 3 NIMMER, supra note 1, § 13.03. Thus, the courts have developed a two prong test, termed the substantial similarity test, to help the trier of fact answer both these questions. Id.

The first prong of the substantial similarity test is termed the “extrinsic” test. The “extrinsic” test requires the trier of fact to “dissect” the two works in controversy and utilize lay and expert testimony to determine whether the similarities between the two works are (a) sufficient to prove copying, if there is some evidence of access, or (b) so striking as to preclude the possibility that the two works were created independently, if there is no evidence of access. Arnstein v. Porter, 154 F.2d 464, 468 (2d Cir. 1946), cert. denied, 330 U.S. 851 (1947).

The second prong is termed the “intrinsic” test. The “intrinsic” test is used only if the “extrinsic” test has established copying. Id. In the “intrinsic” test, only the responses of the lay observer are employed to determine whether the copying was illicit under the Copyright Act. Id.

15. See infra notes 35-41 and the accompanying text discussing nonliteral component copyrightability.

16. See infra notes 42-46 and the accompanying text discussing nonliteral component substantial similarity.

17. The basis underlying the copyright law is that the law protects the expression of an idea but not the idea itself. Computer Assocs., 982 F.2d at 703 (referencing Baker v. Selden, 101 U.S. 99 (1879)). The basis appears straight forward, but, even after thirty years of evolving several aspects of the copyright infringement analysis, Judge Learned Hand still concluded that, “Obviously, no principle can be stated as to when an imitator has gone beyond copying the ‘idea,’ and has borrowed its ‘expression.’ Decisions must therefore inevitably be ad hoc.” Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487, 488 (2d Cir. 1960).

18. See infra notes 79-82 and the accompanying text discussing the various court de-
the courts utilize five statutory tools to interpret the proper scope of protection for the computer program nonliteral components: (1) the scope of protection as it is defined by the nature of a computer program;\textsuperscript{19} (2) the scope of protection as it is defined by prior case law;\textsuperscript{20}

\textsuperscript{19} Developed tests employed to analyze the computer program copyright nonliteral infringement issue.

19. A basic understanding of (1) the elements of a computer and the relationships that the elements of a computer have with each other, and (2) the role that a computer program serves in those relationships, allows the most computer illiterate person to comprehend the nature of a computer program. See G. Gervaise Davis III, Esq., Software Protection 8-17 (Van Nostrand Reinhold Co., Ltd. 1985).

A computer traditionally contains four elements: (1) a central processing unit (CPU)—the "thinking" part of the computer that does computations; (2) the computer’s memory—an area for storing computer programs and other information; (3) an input device to get information into the CPU for doing computations—the keyboard or another device; (4) an output device—a viewing monitor, screen or a printer. Davis, supra note 5, at 302-03.

A CPU is really just a big calculator that utilizes electronic impulses to perform various functions. The various functions that the CPU performs are determined by the computer programs that "tell" the CPU how to "think". The program knows what to tell the CPU in a given situation because a person, called a computer programmer, wrote a set of instructions telling the CPU what to do in certain situations. See Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 550 (E.D.N.Y. 1991), aff'd, 982 F.2d 693 (2d Cir. 1992).

In order for the programmer to understand and devise how a program must control the CPU, the programmer determines first the ultimate task the program will perform. Davis, supra note 5, at 304-07. Then, the programmer breaks down the ultimate task into sub-tasks, and sub-sub-tasks, and etc., as is necessary to utilize the basic arithmetic functions the CPU can perform. \textit{Id.}

The sub-tasks and sub-sub-tasks that all help perform a related function are called components. Nimmer, supra note 1, § 13.03[F][1]. There are several components that go into writing a computer program. \textit{Id.} These components categorize the ideas in the program, and the expression of an idea is the particular way the component functions to achieve the result orientated idea. \textit{Id.}

For this comment, understanding the particular nature of each component is not necessary. However, understanding that the components interact with each other when the CPU operates, is necessary. The components appear static and independent when they are written in code on paper. In operation though, the components are very active and they "behave". See Computer Assocs., 775 F. Supp. at 560.

To visualize how the components in a computer program “behave,” think of the computer program as a recipe for baking bread. In text form, the recipe is nothing more than instructions for combining various ingredients. In practice, the recipe is the combination of ingredients in a bread pan in the oven. The bread in the oven is the same combination of ingredients as the ingredients in the text form, but the actual combination actively "behaves" unlike the words in the recipe text. Analogously, the components in the computer program actively “behave” when they are put into operation in the CPU. This “behavior” displays characteristics of the components that the text cannot describe when the text is merely read literally in written form.

The components of the program are broken down into literal and nonliteral components. See Lotus, 740 F. Supp. at 44-46. Since this comment is primarily concerned with the nonliteral comments, this comment will not fully address tangential issues solely involving literal components. The heart of the computer program copyright nonliteral in-
(3) the scope of protection as it is defined by economic considerations;\textsuperscript{21} (4) the scope of protection as it is defined by the Constitution;\textsuperscript{22} and, (5) the scope of protection as it is defined by the Copyright Act.\textsuperscript{23} Of these

fringement issue concerns the “behavior” of the text of the program—that is the components of the program that are part of the text of the program but not literally written in code as such. \textit{See} Computer Assocs., 982 F.2d at 706. The nonliteral components include: misbehavior, macros, data structures, file structures, data flow, control structures, flow charts, problem decomposition, and set of services. Davis, \textit{supra} note 5, at 318-325.

\textsuperscript{20} \textit{See infra} notes 76-82 and the accompanying text discussing the tests the courts have developed to confront this issue.

\textsuperscript{21} Two significant doctrines guide the scope of economic considerations. The first is the natural law approach to economics that promotes the supply and demand theory with a hands-off attitude toward governmental intervention. Menell, \textit{supra} note 1, at 1059. Second, is the United States Constitution. The Constitution specifically grants Congress the power, “to 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.” U.S. CONST. Art. I, § 8, cl. 8. \textit{See} Menell, \textit{supra} note 1, at 1060-61.

Thus, the issue becomes how much protection does the computer program owner need in order to encourage successful development in computer technology, while at the same time benefiting the public good. \textit{Id.} at 1059. The natural law approach says that a good product will fetch a profit and that competition will maintain a fair profit to benefit the public good. \textit{Id.} at 1060-61. The courts have interpreted the constitutional consideration in favor of promoting the public good regardless of whether that means more protection or less protection than the natural law approach dictates. Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975). Consequently, because of the supremacy of the Constitution, “The ultimate aim is, by this incentive [copyright protection], to stimulate artistic creativity for the general public good.” \textit{Id.}

\textsuperscript{22} In \textit{Twentieth Century Music Corp.}, the Supreme Court advised courts on how to establish the proper constitutional standard for the copyright protection of new subject matter as follows:

\[\text{The copyright law seeks to establish adequate equilibrium. On the one hand it affords protection to authors as an incentive to create, and on the other, it must appropriately limit the extent of that protection so as to avoid the effects of monopolistic stagnation. In applying the federal act to new types of cases, courts must always keep this symmetry in mind.}\]

\textit{Computer Assocs.}, 982 F.2d at 696 (quoting \textit{Twentieth Century Music Corp.}, 422 U.S. at 156).

\textsuperscript{23} \textit{See supra} notes 2-6 and the accompanying text discussing the scope of copyright protection the Copyright Act grants to computer subject matter.

Further defining this scope of copyright protection is the influence of the “legislative history” behind the Copyright Act. \textit{Whelan}, 797 F.2d at 1240-42. The literal text of the Copyright Act does not include the components of a computer program that the Copyright Act intends to protect. 17 U.S.C. §§ 101, 117. Rather, the Copyright Act defines a computer program in section 101, and then the Copyright Act discusses the scope of protection for a computer program in section 117. \textit{Computer Assocs.}, 982 F.2d at 702.

The “legislative history,” however, suggests that copyright protection extends beyond the literal code of the program to include the literal and nonliteral components. \textit{Id.} Congress adopted as “legislative history,” almost verbatim the report of the Commission on New Technological Uses of Copyrighted Works (CONTU)—the commission appointed by Congress to study and report on the problems and interactions between technology and
five tools, the first tool is the most important for "determining" the proper scope of nonliteral protection, because the law cannot protect that which it does not understand.

This comment first discusses two cases, setting up the background necessary to understand why a uniform analysis is essential for determining computer program copyright nonliteral infringement. Second, this comment explains why the tests developed by the federal courts fail to resolve the computer program copyright nonliteral infringement issue. Third, this comment develops the bifurcated uniform analysis the courts should adopt to determine the proper scope of copyright protection for the nonliteral components in a computer program. Finally, this comment demonstrates that the bifurcated uniform analysis integrates both the practical and the theoretical aspects of copyright law. Now, only if the courts unite and adopt the bifurcated uniform analysis, will the computer industry be able to continue writing an "expressive" future.

II. BACKGROUND

A. THE *WHELAN ASSOCIATES v. COMPUTER ASSOCIATES* BATTLE

Computer program nonliteral components were first protected in the computer program infringement case of *Whelan Associates v. Jaslow Dental Laboratory.* Now, seven years after that decision, only five of the thirteen federal circuits have formulated their own opinions about this controversial decision. However, not until 1992, in the case of [ citation needed], only five of the thirteen federal circuits have formulated their own opinions about this controversial decision. However, not until 1992, in the case

COPYRIGHT LAW. *Whelan*, 797 F.2d at 1241 (quoting CONTU Final Report 21 (1978)). CONTU advised Congress that, "The separation of the idea from Form of expression . . . is better realized through the courts exercising their judgement in particular cases [than by a per se rule]. . . . Flow charts, source codes, and object codes are works of authorship in which copyright subsists." *Id.*

24. This word is used very loosely in this comment, precisely because of the illusive nature of the nonliteral components in a computer program. *See Computer Assocs.*, 982 F.2d at 706. A uniform analysis will not be able to decisively determine what "is" the protectable expression in a computer program, but rather, one must determine what "is not" protectable expression in the process of determining whether there has been nonliteral component infringement. *See infra* notes 67-75 and the accompanying text discussing the exclusionary tests the courts use to define what is not protectable expression.

25. *See 3 NIMMER, supra* note 1, § 13.03[F].

26. Only the expression of an idea is protectable. *See supra* note 17 discussing this proposition. Therefore, as long as the computer industry cannot uniformly determine what components of the computer program are protectable in court, software pirates will continue to rob the computer industry of its profits and pirating may eventually put the computer industry out business in the open market. *See How Bad Is The Software Problem?, 5 SOFTWARE L. BULLETIN, March/April 1992, at 50.

27. CARY H. SHERMAN, COMPUTER SOFTWARE PROTECTION LAW § 203.7(b) (1991).

28. *See Whelan*, 797 F.2d at 1248 (3d Cir. 1986); Plains Cotton Co-op Assoc. v. Goodpasture Computer Service, Inc., 807 F.2d 1256, 1262 (5th Cir. 1987), *reh'y and reh'y en banc*
of Computer Associates International v. Altai, Inc.\textsuperscript{29} did a court develop a solid analysis to rival the Whelan analysis.\textsuperscript{30} By evaluating Whelan and Computer Associates, one will understand society’s need for a uniform analysis, and one will begin to see the elements that must be included in a uniform analysis.

B. THE WHELAN ASSOCIATES CASE

In Whelan Associates v. Jaslow Dental Laboratory,\textsuperscript{31} a panel of three circuit judges held that a computer program’s nonliteral components were copyrightable.\textsuperscript{32} In reviewing the district court decision, the circuit court first discussed the issue of copyrightability to determine whether the plaintiff’s computer program nonliteral components were copyrightable subject matter within the Act. This issue invoked the ever developing doctrine of the idea/expression dichotomy.\textsuperscript{33} This dichotomy required the court to distinguish between the ideas in a com-

\textsuperscript{29} Computer Assocs., 982 F.2d 693 (2d Cir. 1992).
\textsuperscript{31} Whelan, 797 F.2d at 1222.
\textsuperscript{32} Whelan Associates (Whelan) was the developer of a custom computer program, Dentalab. Dentalab was designed for Jaslow’s dental prosthetics and devices manufacturing business. Jaslow Dental Lab. (Jaslow) licensed Dentalab from Whelan. \textit{Id.} at 1225.

Two years after Jaslow had been using the program, Jaslow developed its own program, Dentacom. Dentacom performed the same functions as Dentalab. No longer needing Whelan’s services, Jaslow severed its relationship with Whelan. Further, Jaslow recommended that Whelan stop marketing Dentalab because Dentalab revealed Jaslow trade secrets. \textit{Id.} at 1226.


The district court agreed with Whelan, and the court of appeals affirmed the district court decision. \textit{Whelan}, 797 F.2d at 1248.

\textsuperscript{33} The idea/expression dichotomy arises out of the “fundamental principle of copyright law that a copyright does not protect an idea, but only the expression of the idea.” Computer Assocs., 982 F.2d at 703-704.
puter program and the expression of those ideas, because only the expression of the ideas was copyrightable. To respect this dichotomy, the court formulated the broad rule that, "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" for determining copyrightability. The rule "drew a line" defining the purpose or function of the program as the idea in the program, and everything else not necessary to that idea as the expression in the program. The court next considered the opinions of computer experts testifying about the nature of the computer programs in controversy. Then, as a matter of law, though acknowledging the expert testimony, the court employed its own lay observations to define the purpose of the plaintiff's program as, "an aid in the business operations of a dental laboratory." Finally, the court adopted the district court's definition of the expression of the idea as the sequence, order, and structure of the plaintiff's program.

Next, in reviewing the district court's decision, the circuit court discussed the district court's finding of substantial similarity. The district court, as the finder of fact, resolved the substantial similarity question. The district court employed its own lay observations while also relying on the expert testimony it had heard previously. In *Whelan*,

---

34. This principle is codified in the Copyright Act as follows: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method, of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b) (1988).

35. *Whelan*, 797 F.2d at 1236. See also *infra* notes 98-104 and the accompanying text discussing why this rule is over broad.

36. Furthermore, one court has even defined the standard of a computer literate lay observer as follows: "In the context of computer programs, the 'ordinary reasonable person' with the ability to intelligently respond to computer expression is a computer programmer." *Atari*, 975 F.2d at 844 (citing Johnson Controls v. Phoenix Control Sys., 886 F.2d 1173, 1176 n.4 (9th Cir. 1991)).


38. *Whelan* 797 F.2d at 1238.


40. The finder of fact can be either the court or the jury. *See Manufacturers Tech. v. Cam, Inc.*, 706 F. Supp. 984, 1000 (D. Conn. 1989). *See supra* note 14 discussing the two-prong substantial similarity inquiry.

41. Under the traditional bifurcated substantial similarity analysis, the finder of fact could not rely on the expert testimony admitted for determining the scope of copyright
the circuit court affirmed the district court’s finding that the defendant’s Dentacom program was substantially similar to the plaintiff’s Dentalab program. Therefore, the Whelan court was led to also affirm the district court’s finding of infringement of the plaintiff’s program. The Whelan decision became the first federal circuit court of appeals case to find computer program copyright nonliteral infringement.

The Whelan analysis adopted one essential element necessary in a uniform analysis, the new single-prong substantial similarity inquiry. Conversely, most prior cases employed a two-prong substantial similarity inquiry. However, the Whelan analysis also contained three critical mistakes, all regarding the nature of a computer program, which destroyed any further precedential value the holding might have contained. One case strongly criticizing the Whelan decision is Computer Associates International v. Altai, Inc.

C. THE COMPUTER ASSOCIATES CASE

In Computer Associates International v. Altai, Inc., the court was confronted with the same copyrightability task that the Whelan court had “easily” completed by defining the idea in the infringed computer program at issue. Recognizing the inherent flaw in making a rule to “draw a line” between an idea and the expression of the idea, the Computer Associates adopted the Whelan single-prong substantial similarity question, Id. at 713, and Computer Associates also adopted the Whelan syllogism that gives nonliteral components copyright protection. Id. at 702.

Computer Associates could adopt only these two contributions from Whelan because, “[u]nfortunately, the simplicity that makes the [Whelan] test so attractive, also makes it ‘conceptually overbroad’ and descriptively inadequate.” CMAX/Cleveland, Inc. v. UCR, Inc., 804 F. Supp. 337, 352 (M.D. Ga. 1992).
puter Associates court declined to develop such an "easy" rule. Rather, the court established an exclusionary test by defining what is not copyrightable expression. More significantly, the court then formulated a new analysis, by integrating several of the existing exclusionary tests to try and uniformly determine the computer program copyright nonliteral infringement issues.

The new analysis contained three steps. The court employed these three steps to address the issues of copyrightability and substantial similarity in a structured analysis, respecting the nature of a computer program. In the first step, the court subjected the infringed program to the "abstraction" test, separating the literal components from the nonliteral components, and the various nonliteral components from each other. In the second step, the court applied the "filtration" test to the

---

49. See Computer Assocs., 775 F. Supp. at 559-60 (discussing that the nature of a computer program makes the Whelan rule inherently flawed).

50. The court takes out of the Whelan decision that separating idea from expression should not rely too heavily on metaphysical distinctions. Computer Assocs., 982 F.2d at 706. A computer program is more than just textual instructions so a proper analysis of the nonliteral infringement issue must place enough emphasis on the practical considerations of the program, as it is both text and behavior. Id.

51. See infra notes 54-64 and the accompanying text for a discussion of the Computer Assocs. analysis.

52. Another court has recognized the importance that determining copyrightability as a matter of law makes in the overall determination of infringement:

While the issue of whether there has been an infringement ... is made properly on a case-by-case basis, what typically also occurs in the case law analysis is that the determination of what is protectable ... is also undertaken on a case-by-case basis, often with little attention given to the distinction between determining what is protected and what is an infringement.


53. The first two steps in this analysis involve the issue of copyrightability. Therefore, the court as a matter of law should resolve this issue. See notes 4 and 54 setting forth this proposition.

54. Computer Assocs. adopts the Nichols abstraction test, but Computer Assocs. molds it to "fit" the nature of a computer program:

As an anatomical guide to this procedure the following description is helpful: At the lowest level of abstraction, a computer program may be thought of in its entirety as a set of individual instructions organized into a hierarchy of modules. At a higher level of abstraction, the instructions in the lowest level modules may be replaced conceptually by the functions of those modules. At progressively higher levels of abstraction, the functions of higher level modules conceptually replace the implementations of those modules in terms of lower level modules and instructions, until finally, one is left with nothing but the ultimate function of the program. ... A program has structure at every level of abstraction at which it is viewed. At low levels of abstraction, a program's structure may be quite complex; at the highest level it is trivial.

Computer Assocs. 982 F.2d at 707. (quoting Steven R. Englund, Note, Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs, 88 MICH. L. REV. 866, 897-98 (1990)).
allegedly infringed program, as follows: 55 First, the court defined a non-copyrightable component to be one that fits the definition of an exclusionary test. 56 Second, the court eliminated from this infringement analysis all the non-copyrightable components, because only copyrightable components of the plaintiff's program can be infringed. 57 In the third step of the new analysis, the court gathered the remaining components, the copyrightable components. 58 Then, the finder of fact compared the copyrightable components in the allegedly infringed program with the similar components in the allegedly infringing program. 59 Finally, the court determined that the district court's finding of no substantial similarity between the two programs was not clearly erroneous. 60 Thus, the court was obligated to affirm the district court's denial of relief to the plaintiff. 61 Furthermore, the court commented that, "so long as the appropriated material consists of non-protectable expression, this result is neither unfair nor unfortunate." 62

---

55. See generally 3 NIMMER, supra note 1, § 13.03[F] discussing the filtration procedure.

56. The filtration procedure is applied to the infringed work and not the infringing work, for two reasons. First, because, whether there exist copyrightable components in the infringing work is irrelevant to whether the defendant's program infringes on the plaintiff's copyrightable components. Thus, the filtration of the defendant's program "would be wasteful and unnecessarily time consuming." Computer Assocs., 982 F.2d at 714. Second, "[B]y focusing the analysis on the infringing rather than on the infringed material, a court may mistakenly place too little emphasis on a quantitatively small misappropriation which is, in reality, a qualitatively vital aspect of the plaintiff's protectable expression." Id.

60. See infra notes 67-75 and the accompanying text for a discussion of the exclusionary tests.

61. Id. at 715.
II. ANALYSIS

A. MOVING TOWARDS A UNIFORM ANALYSIS

Throughout the history of copyright nonliteral infringement cases, the federal courts have utilized two groups of tests to help them resolve the nonliteral infringement issues. Separately, these tests have failed to resolve the computer program copyright nonliteral infringement issues. United, however, these tests significantly define the substance of a uniform analysis. Thus, a uniform analysis based on these tests will be an analysis that properly "determines" the scope of copyright protection granted to the nonliteral components in a computer program.

The first group of tests are called the exclusionary tests. The courts never intended for these tests to stand on their own to create a uniform analysis for copyright infringement analyses. Rather, these tests are elements to be used in determining subject matter copyrightability. These tests are divided into two sub-groups depending on their origin in copyright law. The first sub-group includes the exclusionary doctrines that have developed from the general case history of copyright law. These doctrines are grounded in the fundamental principles of copyright law as interpreted from the Constitution. The three most significant exclusionary doctrines are the merger doctrine. Because ideas are not always separable from their expression, the merger doctrine exists to "give the benefit of the doubt" to protecting the public and thus prohibits the Copyright Act from protecting ideas that may be so close to their expression as to be effectively inseparable. "[W]hen the 'idea' and its expression are thus inseparable, copying the expression will not be barred, since protecting the expression would confer a monopoly of the 'idea' upon the copyright owner free of the conditions and limitations imposed by the patent law." Gates, 798 F. Supp. at 1517 (quoting Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971)).
the scenes a faire doctrine,71 and the public domain doctrine.72 The second sub-group includes the exclusionary rules that have developed from the text of the current copyright statute.73 The courts apply these two sub-groups of tests in each case involving copyrightable subject matter issues because these tests define the basic scope of copyright protection.74

The second group of tests are called the court-developed tests. The courts develop these tests to stand on their own as a uniform analysis for determining computer program copyright nonliteral infringement; specifically, the substantial similarity inquiry.75 The courts derive these tests from the Copyright Act by paying particular attention to interpreting copyright law in light of the nature of a computer program.76 The court-developed tests do not yet carry the legal significance that the exclusionary tests do because these tests are relatively new and do not yet reflect the view of the majority of the courts.77

Copyright case law recognizes the following four court-developed

dataion is whether the use of a particular component is the only, or one of a few, means to efficiently implement a particular idea in the program. Computer Assocs., 982 F.2d at 708.

71. The doctrine of scenes a faire prohibits copyright protection of expression that naturally follows from the author's work, without the author having intended to create that expression. Whelan, 797 F.2d at 1236.

In relation to computer program components, this doctrine includes the external considerations that limit the number of efficient ways a program can be designed to perform a particular task on a particular computer. Computer Assocs., 982 F.2d at 709-10. These external considerations include: CPU standards, program standards, manufacturing standards, target industry practices, and computer industry programming practices. 3 Nimmer, supra note 1, § 13.03[F][3].

72. Material that is in the public domain is free for everyone to use equally. Computer Assocs., 982 F.2d at 710.

In the computer program context, allowing program owners to copyright program components already existing for public use cannot be protected because the owner did not create this material with the Constitutional minimum of creativity in "their [own] respective writings." U.S. Const. art. I, § 8, cl. 8.

Furthermore, such a grant of copyright protection would offend both the constitutional text as well as the constitutional concept to promote the public good through computer technology. 3 Nimmer, supra note 1, § 13.03[F][4].

73. The current copyright statute is the 1988 Copyright Act. 17 U.S.C. §§ 101-914 (1988). In the Act, sections 107 - 120 list the exclusionary rules that apply either to all copyrightable subject matter, e.g., independent creation, fair use, etc., or to particular subject matter, e.g., literary works, musical works, computer programs, etc. 17 U.S.C. §§ 107-120 (1988).

74. See generally Brown Bag Software v. Symantec Corp., 960 F.2d 1465, 1475 (9th Cir. 1992), cert denied, 113 S. Ct. 198 (1992) (utilizing the filtration step to separate protectable computer program components away from unprotectable ones).

75. See 3 Nimmer, supra note 1, § 13.03[A]. See also infra notes 83-123 and the accompanying text discussing these court developed tests.


tests. The first test is the *E.F. Johnson Co. v. Uniden Corp.* "iterative" test. The second test is the *Whelan* "structure, sequence, and organization" test. The third test is the *Broderbund Software v. Unison World* "total concept and feel" test. Finally, the fourth test is the *Nichols v. Universal Pictures Co.* "abstraction" test, applied in the most recent case on this issue, *Computer Associates International v. Altai, Inc.* Although the court-developed tests were intended to stand on their own and answer the substantial similarity question, these tests have failed to fulfill that role in copyright law.

**B. DEFICIENCIES IN THE COURT DEVELOPED TESTS**

1. **The "Iterative" Test**

The first computer program case to develop a nonliteral infringement test was *E.F. Johnson Co. v. Uniden Corp.* Essentially, the court developed a two-part "iterative" analysis. The court restated the first-prong of the traditional substantial similarity question in terms of whether the defendant "used" the plaintiff's program to develop his own. Then, the court defined a new test to replace the second-prong of the traditional substantial similarity question. The new test asked whether the defendant's program was an iterative reproduction of the plaintiff's program, "that is, one produced by iterative or exact duplication of substantial portions of the copyrighted work."

Although *E.F. Johnson* contributes one positive element toward a uniform analysis, its focus on utilizing expert testimony in the second part of the iterative analysis, the court fails to resolve three other problems. First, the "use" standard the court applies, in the first part of

---

82. See 3 NIMMER, supra note 1, § 13.03[A][1][d].
83. The following two-part analysis was developed to confront the computer program copyright nonliteral infringement issue:

The iterative approach requires proof (1) that the defendant "used" the copyrighted work in preparing the alleged copy, which may be established by proof of access and similarity sufficient to reasonably infer use of the copyrighted work; and (2) that the defendant's work is an iterative reproduction, that is, one produced by iterative or exact duplication of substantial portions of the copyrighted work.

---

84. 3 NIMMER, supra note 1, § 13.03[A][1][d].
85. In the second part of the iterative test, the focus is on the court's evaluation of whether similarities are illicit. See supra note 14 discussing illicit copying. Thus, the court should utilize its own lay opinion, in conjunction with expert opinion to determine whether the quality and the quantity of similarities suggest the defendant unlawfully copied the plaintiff's program. *E.F. Johnson*, 623 F. Supp. at 1493.
the iterative analysis, relies on some degree of similarity to prove copying by access\(^8\), but the court does not explain how much similarity is necessary.\(^7\) Second, the iterative reproduction test also fails to explain what level of similarity is necessary for a finding of infringement, other than in the cases of exact duplication\(^8\) or iterative duplication. The court never explains what is iterative duplication.\(^9\) However, one could presume it means the defendant blatantly copies components of the plaintiff’s program that no longer serve any function in the working program.\(^9\) Third, even if the defendant did “use” the plaintiff’s program, i.e., if the defendant only took ideas from the plaintiff’s work, then there is no infringement since infringement requires illicit copying.\(^9\)

2. The “Structure, Sequence, and Organization” Test

*Whelan Associates v. Jaslow Dental Laboratory* made the first “good” attempt to develop a uniform analysis for confronting the computer program copyright nonliteral infringement issue.\(^9\) The court addressed the idea/expression problem head-on by defining what is an idea.\(^9\) Then, it applied this definition to any plaintiff’s program to dis-

---

\(^8\) Access can be proven by showing that the allegedly infringed program was for sale on the open market, and that plaintiff could have purchased a copy of the program. *Computer Assocs.*, 775 F. Supp. at 558.

\(^7\) See 3 Nimmer, *supra* note 1, § 13.03[A][1][d].

\(^8\) In other words literal copying, which would lead a court to apply the traditional literal infringement analysis. See *supra* note 14 discussing the literal infringement analysis.


\(^9\) See 3 Nimmer, *supra* note 1, § 13.03[A][1][d].

As a program is revised and updated, often components no longer serve a working function in the program. However, a programmer may leave these components in the text of the program, because they do not harm the program either. Usually, an illicit copier will not figure out that these components serve no function and thus verbatim copying of these useless components suggests illicit copying. Davis, *supra* note 5, at 318-19.

\(^9\) Professor Nimmer defines the place the filtration step should be utilized in the nonliteral infringement analysis based on the premise that only copyrightable expression is protectable: “Infringement is shown by a substantial similarity of protectable expression, not just an overall similarity between the works. Thus, before evaluating substantial similarity, it is necessary to eliminate from consideration those elements of a program that are not protected by copyright.” See 3 Nimmer, *supra* note 1, § 13.03[F].

\(^9\) See *supra* notes 30-52 discussing the elements in *Computer Assocs.*, adopted from the *Whelan* analysis.

\(^9\) See *supra* note 35 discussing the idea/expression dichotomy.

\(^9\) “[T]he 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.” *Whelan*, 797 F.2d at 1236. See also *supra* notes 31-64 and the accompanying text discussing *Whelan*. 
tistinguish the expression from the idea in a component. Consequently, since the expression is the protectable part of the component, this protectable part could then "easily" be compared to the similar expression in the defendant's program. Finally, the court, as finder of fact, would utilize expert testimony to determine if there was enough similarity to rise to the level of illicit copying.

Unfortunately, the Whelan court made three mistakes in its analysis, all stemming from the court's misunderstanding of computer technology. First, flowing directly from the definition distinguishing idea from expression, is the broad conclusion that all that is not idea, is the expression of an idea. This conclusion is inherently flawed because of the nature of a computer program as a combination of ideas, expression, and a mixture of ideas and expression. Second, the court used the terms, "structure, sequence, and organization" interchangeably in their analysis, where all three words were understood to be synonymous. The problem with this view is that technically these terms are not synonymous. Technically, the structure defines the internal ordering of the components source and object code; the organization defines the static relationships between the components in a program and the ordering of the components as a whole; and, the sequence defines the active organizational relationships, the "behavior," between the components when the computer program is in operation. Furthermore, an infringement test finding these three terms equivalent is fundamentally flawed because the test does not understand the subject

95. Johnson Controls, 886 F.2d at 1175.
96. See notes 98-104 and the accompanying text discussing why this task is not so "easy."
97. See also infra notes 101-102 and the accompanying text discussing the inherent flaws in the Whelan rule.
98. The Whelan court adopted the E.F. Johnson single-prong substantial similarity test, whereby the court utilized expert testimony in making both determinations of copying by access and illicit copying. Whelan, 797 F.2d at 1232-33.
99. See 3 Nimmer, supra note 1, § 13.03[F].
100. A computer program's ultimate function or purpose is the composite result of interacting components, where each component may be purely an idea or the mixture of idea and expression. Computer Assocs., 982 F.2d at 705.
101. In the opening paragraph of its decision, the court dropped footnote one: "We use the terms 'structure,' 'sequence,' and 'organization' interchangeably when referring to computer programs, and we intend them to be synonymous in this opinion." Whelan, 797 F.2d at 1224 n.1.
102. See supra note 101 supporting this explanation.
Third, the Whelan decision failed to recognize that programs may have more than one idea. In fact, computer programs are based on numerous ideas that interplay with each other at the various levels the program operates. If Whelan has taught the legal community anything new, it is that a copyright nonliteral infringement analysis cannot include a bright-line test to distinguish ideas from expression. Rather, the analysis must begin by recognizing the multi-dependent component nature of a computer program, and build on that premise.

3. The "Total Concept And Feel" Test

The "total concept and feel" test originated outside the computer world. This test was first adopted in a computer program context in the case of Broderbund Software v. Unison World. The court utilized the total concept and feel test to help the ordinary lay observer determine the second prong of the traditional substantial similarity question, the intrinsic test. The court articulated the intrinsic test as follows: "The finder of fact . . . is to determine whether an 'ordinary reasonable person' would find the expression of the subject works to be substantially similar. The question is whether the infringing work captures the 'total concept and feel' of the protected work." Then, the court as finder of fact, found substantial similarity and thus infringement, because, "[t]he ordinary observer could hardly avoid being struck by the eerie resemblance," between the two programs.

The Broderbund test fails to provide a uniform analysis to the copyright nonliteral infringement issue, for two reasons. First, by employing the "total concept and feel" test to answer the second prong of the substantial similarity question, the intrinsic test, the court violates the fundamental principle of copyright law, which is, ideas are not copyrightable. The "total concept and feel" test more closely defines an idea in a work than the expression of the idea. The "total concept and feel" test was first developed in a pictorial works case involving the copyright infringement of greeting cards.

In the pictorial works context, the whole work can be viewed and

103. Davis, supra note 5, at 314.
104. Computer Assocs., 982 F.2d at 705.
106. E.F. Johnson, 623 F. Supp. at 1492-93 (quoting Roth Greeting Cards v. United Card Co., 429 F.2d 1106, 1110 (9th Cir. 1970)).
107. 3 NIMMER, supra note 1, § 13.03[A][1][c].
109. Id.
110. 3 NIMMER, supra note 1, § 13.03[A][1][c].
111. Roth Greeting Cards v. United Card Co., 429 F.2d 1106, 1110 (9th Cir. 1970).
understood in a single glance. The "result" of a pictorial work, viewed by a lay observer, is the work's expression. On the other hand, the expression in a computer program is the way a program achieves a particular result or purpose, not the actual result itself, which is usually an idea. A lay observer often does not understand the nature of a computer program, let alone be able to distinguish between the ideas and the expression of a program. Therefore, when a lay observer is asked to determine the similarity between two programs based on the total concept and feel, the observer can only compare what she understands—that is the result or the purpose of the programs, the ideas. Thus, the "total concept and feel" test wrongly encourages the lay observer to determine similarity between programs, as based on the ideas the programs contain, when the lay observer should be basing similarity on the expression of the ideas as the proper subject matter of copyright protection.

Second, this test loses all context when dealing with complex subject matter, such as computer programs. The only way the "total concept and feel" test may be justified is if it is limited to a class of "ordinary observers" that understand the computer subject matter in controversy. However, since highly computer literate individuals are not yet a majority of society, common sense dictates that the first prong of the substantial similarity question, the extrinsic test, should determine the substantial similarity question.

4. The "Abstraction" Test

Computer Associates International v. Altai, Inc., is the first computer program copyright nonliteral infringement case to formally adopt the abstraction test in a program infringement analysis. Applying
the "abstraction" test,120 the Computer Associates court derived the nonliteral components in a computer program from the allegedly infringed program.121 Accepting the premise that no bright-line can be found to formulate a general rule distinguishing idea from expression,122 the court molded the abstraction test to "fit" a computer program.123

In itself, the utility of the abstraction test ends here.124 The abstraction test does not directly help the finder of fact determine the nonliteral infringement issues, although it does aid the finder of fact in this task.125 The creator of the test, Judge Learned Hand, even found that after applying the abstraction test, the infringement analysis becomes "ad hoc."126 Furthermore, the infringement analysis involves more than a determination of copyrightability, it also involves the determination of substantial similarity.127 Consequently, the abstraction

---

120. Here is the original "abstraction" test as Judge Learned Hand applied it to a theater play:

[U]pon 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. 

Nichols, 45 F.2d at 121.

121. Computer Assocs., 982 F.2d at 706-07. See supra note 54 and the accompanying text setting forth exactly how this court applied the abstraction test to the allegedly infringed program.

122. Id. at 704 (quoting Peter Pan Fabrics v. Martin Weiner Corp., 274 F.2d 487, 489 (2d Cir. 1960)).

123. The court did not have to evolve the abstraction too much because a computer program is developed in levels, as is a theater play for which the "abstraction" test was originally developed. Gates, 798 F. Supp. at 1512-13.

Furthermore, a computer program is designed in a "top down" fashion. "Top down" means that the programmer starts with the end purpose or function that she wants a computer program to reach, e.g., adding two numbers, or word processing. Then she designs a flow chart, beginning at the top with the purpose or function to be reached by the program, and then breaking down the purpose into sub-purposes and sub-sub-purposes that will help fulfill the end purpose. 3 Nimmer, supra note 1, § 13.03[F][1].

Thus, starting with the most basic program function, e.g., a word processing program, or a spreadsheet program, the programmer designs "down" to the most complex level, e.g., the electronic impulses that the CPU works with in its memory. Id.

124. Id.

125. See infra note 132 and the accompanying text discussing the utility of the abstraction test.

126. Peter Pan Fabrics, 274 F.2d at 489.

127. See infra notes 129-140 and the accompanying text discussing all the elements necessary in a uniform analysis.
test gives the court no guidance in determining how much similarity should constitute substantial similarity, so the abstraction test cannot be its own uniform analysis.\textsuperscript{128}

5. \textit{What Do These Tests Mean?}

First, the "abstraction" test breaks down the complex computer program into components that an analysis can more readily utilize to answer the copyrightability and the substantial similarity issues. Second, the "structure, sequence, and organization" test directs the analysis away from drawing a bright-line between idea and expression because of the "behavioral" nature of a computer program. Third, the exclusionary doctrines and rules provide a basis in copyright law from which copyrightable expression can be separated from uncopyrightable expression and ideas, without relying on a bright-line test. Fourth, the "iterative" test suggests utilizing a single-prong substantial similarity question, comprising of only the extrinsic test. Finally, the "total concept and feel" test supports utilizing a single-prong substantial similarity question because of the technical complexities encountered in the field of computer programming. Consequently, despite the positive elements that these tests do contribute to a uniform analysis, no single test provides uniform guidance for determining how similar is substantially similar.\textsuperscript{129} Fortunately, this conclusion does not mean that a uniform analysis cannot provide for a more certain determination of the substantial similarity question than currently exists in the individual court developed tests.\textsuperscript{130}

Upon compiling the elements from all the exclusionary tests, one is led to two conclusions. First, the issue of computer program copyright nonliteral component copyrightability can be answered almost conclusively on any set of facts.\textsuperscript{131} Second, the issue of computer program copyright nonliteral component substantial similarity can be answered with a fair degree of conclusiveness, though not near the degree of conclusiveness as in the copyrightability issue.\textsuperscript{132}

\textsuperscript{128} See 3 NIMMER, \textit{supra} note 1, § 13.03[A].

\textsuperscript{129} First, the exclusionary tests were never intended to provide the court with a uniform analysis of this issue. See \textit{supra} note 67. Second, the prior analysis of the deficiencies in the court developed tests demonstrates that these tests will not properly resolve this issue. See \textit{supra} notes 76-127 and the accompanying text supporting this proposition.

\textsuperscript{130} See \textit{infra} note 142 supporting this proposition.

\textsuperscript{131} The abstraction step and the filtration step determine the copyrightability of components. See \textit{Lotus}, 799 F. Supp. at 215-16. "Once the program's abstraction levels have been discovered, the [copyrightability] inquiry moves from the conceptual to the concrete." \textit{Computer Assocs.}, 982 F.2d at 707.

\textsuperscript{132} See notes 126, 142 supporting this proposition.

Furthermore, although much of the uncertainty can be eliminated by a uniform anal-
C. THE BIFURCATED UNIFORM ANALYSIS

Just as there must be an idea before there can be an expression of the idea, the courts must “define” computer program copyrightable expression before they can determine if there has been an infringement of that expression. Thus, after a court determines that the owner of the allegedly infringed program has a valid copyright, the court and then the trier of fact will make the following two additional determinations. First, a court must determine whether the nonliteral components in the allegedly infringed program are copyrightable. Second, the finder of fact must determine whether similar components in the two programs are so similar to each other that they are substantially similar.

1. Finding The Copyrightable Components

To “answer” the question of copyrightability, the court must use the following two step process: (1) the court must determine and understand the “abstract” components of the allegedly infringed program; and (2) the court must “filter” out any unprotectable components. In this two step analysis, copyrightability is a question of law. However, the court should utilize expert testimony in conjunction with its own lay perspective.

a. The Abstraction Step

The first step in the copyrightability inquiry is “abstraction.” This step employs the “abstraction” test originally developed by Judge Learned Hand. This step is the best starting point for a uniform analysis because of the utilitarian nature of a computer program. Utilizing expert testimony that explains what went into the making of the allegedly infringed program, the court defines the various levels of nonliteral components in the allegedly infringed program. These components will contain ideas, expression of ideas, and a mixture of both.

\footnotesize

\begin{itemize}
  \item See supra note 33 and the accompanying text for a discussion of the initial steps in the literal infringement analysis.
  \item See supra note 39 supporting this proposition.
  \item See supra note 38, where Whelan strongly advances this position.
  \item See supra note 56, where Computer Assocs. adapts Judge Learned Hand’s abstraction test to “fit” computer programs.
  \item See supra note 1, § 13.03[F].
  \item See supra notes 55-56 and the accompanying text supporting this proposition.
  \item See supra note 99 and the accompanying text.
\end{itemize}
b. **The Filtration Step**

The second step in the copyrightability inquiry is "filtration." Sifting through the levels of abstraction, the court filters away unprotectable expression from protectable expression, utilizing the plethora of exclusionary tests recognized in copyright law. After filtration, the court will be left with a kernel of protectable expression.

2. **Determining Whether Similarity Is Substantial**

To "answer" the question of substantial similarity, the finder of fact must follow three steps. First, the trier of fact must admit as evidence expert testimony discussing the degree of similarity between the kernel of components in the allegedly infringed program and the components in the allegedly infringing program. Second, the finder of fact must evaluate the credibility of the expert testimony. Third, weighing the credible expert testimony, if the finder of fact determines there is similarity such that a computer literate person can conclude by a preponderance of the evidence that the infringer copied the allegedly infringed program, then the alleged infringer has illegally infringed the copied program. Consequently, the infringed program owner is entitled to collect infringement damages as dictated by the Copyright Act.

D. **Evaluating the Bifurcated Uniform Analysis**

A bifurcated uniform analysis is supported in copyright law both practically and theoretically. Applying the bifurcated uniform analysis to a computer program demonstrates the certainty in determining the copyrightability of the nonliteral components for two reasons. First, the abstraction step and the filtration step are objective tests that any computer expert, with the aid of legal counsel in copyright law, can determine almost conclusively, without the aid of the courts. Second, when a case does go to trial, the bifurcated uniform analysis eliminates

---

140. *Computer Assocs.*, 982 F.2d 707-10. See *supra* notes 67-75 and the accompanying text discussing these tests.

141. This kernel represents the only protectable components in the allegedly infringed work that will be compared to similar components in the allegedly infringing program for the trier of fact to determine if they are substantially similar. 3 *Nimmer*, *supra* note 1, § 13.03[F][5].

142. *Whelan*, 797 F.2d at 1232.

143. *Id.* at 1246.

144. *Davis*, *supra* note 5, at 325.

145. See *supra* note 9 discussing the grant of copyright protection.

146. Furthermore, experts can apply the proposed uniform analysis before litigation begins so the parties can more accurately determine how the case should come out if it does in fact go to trial. Thus, more often than not, the losing party will settle the suit to avoid large legal fees and a unfavorable judgment. *Davis*, *supra* note 5, at 325.
much uncertainty from the substantial similarity question in the analysis because computer literate experts determine the degree of similarity between the program components at issue, rather than computer illiterate judges and juries. In addition to the practical benefits of the bifurcated uniform analysis, the five statutory tools also support the bifurcated uniform analysis.

1. The First Tool

The scope of protection as it is defined by the nature of a computer program supports the bifurcated uniform analysis for three reasons. First, the abstraction step properly dissects the computer program in accord with the way a program is created—level by level. Second, the filtration step utilizes the plethora of exclusionary tests and avoids creating a bright-line test that can only inaccurately describe a computer program. Third, an expert can best "determine" the substantial similarity issue. An expert in computer programming understands the nature of a computer program. Thus, an expert will make a proper similarity "determination" based on the behavioral nature of the programs at issue, rather than on the "ad hoc" understanding of the lay court or jury.

2. The Second Tool

The scope of protection as it is defined by prior case law also provides support for the bifurcated uniform analysis in the following three ways. First, the abstraction step is deeply rooted in the doctrines of copyright case law. Second, the filtration step provides the court with tests which have already been developed by prior courts and are thus widely utilized in copyright law. Third, case law has supported the use of expert opinion and lay opinion ever since their paired intro-

---

147. The reasoning for this conclusion is as follows. First, the uniform analysis minimizes the amount of information a court may subject to the substantial similarity question in the uniform analysis, i.e., the filtration step. Second, the uniform analysis determines substantial similarity between similar components on the basis of expert testimony. Therefore, a smaller body of similar components are evaluated by computer literate persons, which significantly reduces the chance that similar components will be evaluated on an irrelevant basis. Id.

148. See notes 19-25 and the accompanying text for the discussion of the statutory interpretation tools.

149. Computer Assocs., 982 F.2d at 706-07.

150. See 3 NIMMER, supra note 1, § 13.03[F].

151. Davis, supra note 5, at 325.

152. This test originated in Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), and the test has continued to be adopted through the case of Computer Assocs., 982 F.2d at 706.

153. Thus, future courts can look back on these decisions to help them apply the test
duction in *Arnstein v. Porter.* More significantly, recent case law emphasizes the value in primarily using expert testimony when determining the substantial similarity question.

3. The Third Tool

The scope of protection as it is defined by economic considerations further supports the bifurcated uniform analysis for two reasons. First, the abstraction step and the filtration step can both be performed by each party to a potential law-suit before the case goes to trial, thus, reducing legal costs due to court time and lawyer fees. Second, by utilizing expert testimony, the courts and the parties will not have to waste time and money teaching the finder of fact the intricacies of a computer program.

4. The Fourth And Fifth Tools

The scope of protection, as it is defined by the Constitution and the Act, provides additional support for the bifurcated uniform analysis for three reasons. First, the filtration step is based on the doctrines of copyright law that evolve from the Constitution. Second, the filtration step is also based on the rules in the Copyright Act, so again federal law requires that these tests be applied. Third, the abstraction step and the substantial similarity question do not directly gain support from the Constitution or the Copyright Act. However, since these steps aid the fact finder in the determination of the nonliteral infringement issue, the fourth and the fifth tools indirectly support the bifurcated uniform analysis.

In the computer context. See *supra* notes 69-75 and the accompanying text discussing the exclusionary tests.

154. *Arnstein,* 154 F.2d 464, 468 (2d Cir. 1946).

155. The *Arnstein* two-prong test is well-suited for evaluating works that are readily comprehensible to the lay observer. *Computer Assocs.,* 982 F.2d at 713. However, in the case of computer programs, which are rather impenetrable to judges and juries, the *Arnstein* test must be set aside in favor of the single-prong substantial similarity question, which utilizes expert testimony. *Id.*

156. Thus, the court does not have to waste time developing these tests from scratch. Also, each party will know how to better prepare their case, and may even find it more cost efficient to settle the case. See Menell, *supra* note 1, at 1058.

Furthermore, parties can now dissect their own programs and apply the exclusionary tests to determine if their program contains any copyrightable components, or if they have copied any copyrightable components from another program. See Davis, *supra* note 5, at 325.

157. See 3 *Nimmer,* *supra* note 1, § 13.03[F].

158. U.S. CONST. art. I, sec. 8, cl. 8. (How much more support can a legal analysis get than being partly rooted in the Constitution!)

159. *Computer Assocs.,* 982 F.2d at 711.

160. First, the ultimate aim of copyright law is "to stimulate artistic creativity for the
III. CONCLUSION

The difficulties that Computer Associates encountered in the Whelan decision arose because the Whelan court lacked a basic understanding of computer programs. The analysis that Computer Associates developed was much improved over Whelan because the Computer Associates court laid down a structured analysis incorporating a better understanding of the computer program. However, Computer Associates did not advocate strongly enough the bifurcated need that a uniform analysis would necessarily have to serve in order to properly protect the nonliteral subject matter that a computer program embodies.

The bifurcated uniform analysis clearly recognizes this “bifurcated” need. Building on the Computer Associates computer literate decision, the bifurcated uniform analysis integrates the traditional literal infringement steps, and then, the analysis develops the bifurcated inquiry, first into the copyrightability issues, and second, into the substantiality similarity issues. The bifurcated emphasis makes the bifurcated uniform analysis universally applicable to specifically protecting computer programs nonliteral components. Thus, the bifurcated uniform analysis significantly diminishes the “ad hoc” results the courts have produced through applying the existing analyses in computer program copyright nonliteral infringement cases.

In practice, the bifurcated uniform analysis determines the proper scope of copyright protection for the computer program nonliteral components. More significantly, the statutory tools support adopting the bifurcated uniform analysis. In conclusion, both the courts and the computer industry will benefit from adopting the bifurcated uniform analysis. The bifurcated uniform analysis will produce court opinions that understand, through expert assistance, the subject matter being protected. Thus, the court’s decisions will be uniformly computer-literate for those in the legal profession and the computer programming field to follow.

Michael J. Bendel

general public good.” Twentieth Century Music, 422 U.S. at 156. Second, the proposed analysis will uniformly protect program expression, thus promoting program creativity, i.e., the creation of programs. See supra notes 126-127. Therefore, the Constitution and the Copyright Act support the uniform analysis because it furthers the “ultimate” aim of copyright law.