

## COPYRIGHT PROTECTION: HAS LOOK & FEEL CRASHED?\*

HON. JOHN M. WALKER, JR.

JESSICA D. LITMAN

SUSAN G. BRADEN

ANTHONY L. CLAPES

HENRY B. GUTMAN

ROCHELLE COOPER DREYFUSS

MARCI A. HAMILTON\*\*

---

\* Symposium held March 22, 1993 at the Benjamin N. Cardozo School of Law, New York, New York, under the sponsorship of the *Cardozo Arts & Entertainment Law Journal*.

\*\* *Hon. John M. Walker, Jr.* sits on the United States Court of Appeals for the Second Circuit, a position he has held since 1989. Prior to his current appointment, Judge Walker was named in 1985 as a United States District Judge for the Southern District of New York. Judge Walker authored the court's opinion in *Computer Associates Int'l v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992).

*Jessica D. Litman*, Professor of Law at Wayne State University School of Law in Detroit, Michigan, served as Law Clerk to Judge Betty B. Fletcher of the United States Court of Appeals for the Ninth Circuit from 1983-84. In addition to teaching in the areas of copyright, entertainment contracts, intellectual property, international and comparative copyright, trademarks and unfair competition, and torts, Professor Litman has testified before the House Subcommittee on Intellectual Property & Judicial Administration of the House Judiciary Committee and before the Subcommittee on Patents, Copyrights & Trademarks of the Senate Judiciary Committee.

*Susan G. Braden*, a member of the Washington, D.C. law firm of Ingersol & Bloch, Chartered, concentrates her practice on issues of intellectual property, antitrust, international trade, legislative matters, and complex litigation. Ms. Braden represented the prevailing party, *Altai, Inc.*, in *Computer Assoc's Int'l v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992).

*Anthony L. Clapes* is Assistant General Counsel for International Business Machines Corporation. The author of several works on the subject of protection of computer software, Mr. Clapes is also a frequent speaker on computer law topics.

*Henry B. Gutman* is a partner in the New York firm of O'Sullivan Graev & Karabell, where he has headed the litigation department since joining the firm in 1983. Since 1985, Mr. Gutman has been principal outside litigation counsel for Lotus Development Corporation and served as its lead counsel in *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990).

*Rochelle Cooper Dreyfuss* is Professor of Law at New York University School of Law where her areas of academic concentration include intellectual property and law and science. In addition to her appointment as Law Clerk to Chief Justice Warren E. Burger of the United States Supreme Court from 1982-83, Professor Dreyfuss was Law Clerk to Chief Judge Wilfred Feinberg of the United States Court of Appeals for the Second Circuit from 1981 through 1982.

*Marci A. Hamilton* is Assistant Professor of Law at the Benjamin N. Cardozo School of Law in New York City. Prior joining the faculty at Cardozo, Professor Hamilton served as Judicial Clerk to Justice Sandra Day O'Connor of the United States Supreme Court. From 1988 through 1989, Professor Hamilton served as Judicial Clerk to Judge Edward R. Becker of the United States Court of Appeals for the Third Circuit. At Cardozo, Professor Hamilton concentrates in the areas of copyright law, first amendment law, constitutional law, labor law and jurisprudence.

## I. INTRODUCTION

*Dean Frank J. Macchiarola\*\*\**

The topic of copyright protection for computer software is an important and developing one; and we have been very fortunate to obtain [for this symposium,] some of the most important figures in the field, men and women, who are helping to shape the law.

. . . . I would like to set the stage for these proceedings with some comments about the field itself. The basic issue is grounded in constitutional law: Article I, section 8, clause 8 which frames the basis for copyright protection.<sup>1</sup> The copyright has been given codified protection in furtherance of that constitutional provision. But the purpose of copyright is not to secure an author's proprietary rights. Rather, its purpose is to benefit society by promoting the dissemination of artistic and scientific works. Its mechanism, which provides authors with the incentive to create, is the economic benefit provided by a grant of exclusive rights. The basis for this scheme is "the conviction that encouragement of individual effort by personal gain is the best way to advance the public welfare."<sup>2</sup> Inevitably, however, tensions arise between an author's property interest and the public's interest in access to that property. Those tensions are exacerbated by modern technologies, such as computers, that promote dissemination, but diminish an author's ability to control, and therefore to benefit financially from, the reproduction of their work.

Axiomatic to copyright law is the notion that it will not protect ideas per se,<sup>3</sup> but that expression of ideas, if such expression is minimally original and fixed in a tangible medium of expression, will be afforded protection.<sup>4</sup> Where, however, an idea can

---

\*\*\* *Professor of Law and Dean.* B.A., 1962, L.H.D., 1981, St. Francis College; LL.B., 1965, Ph.D., 1970, Columbia University; LL.D., 1983, Dominican College; L.H.D., 1983, College of Staten Island of the City University of New York; LL.D., 1983, Manhattan College. This introduction was prepared with the assistance of Gary S. Lutzker, 1992-93 Editor-in-Chief, *Cardozo Arts & Entertainment Law Journal*.

<sup>1</sup> U.S. CONST. art. I, § 8, cl. 8.

<sup>2</sup> *Mazer v. Stein*, 347 U.S. 201, 219 (1954).

<sup>3</sup> *Baker v. Selden*, 101 U.S. 99 (1879) was the first case to state this rule. It has since been codified in 17 U.S.C. § 102(b). See *infra* note 4. This is one of the characteristics that distinguish copyright protection from patent protection. "Unlike a patent, a copyright gives no exclusive right to the art disclosed; protection is given only to the expression of the idea not the idea itself." *Mazer v. Stein*, at 217 (citation omitted).

<sup>4</sup> See 17 U.S.C. § 102.

(a) Copyright protection subsists, in accordance with this title, 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 . . . .

only be expressed in one way, or in a very limited number of ways, the expression and the idea are said to “merge,” and copyright protection is forfeited. Similarly, protection is not extended to expression that is necessarily incident or indispensable to the idea.<sup>5</sup> On the other hand, where many various means of expressing an idea are available, the particular means chosen is unnecessary to the idea and is, therefore, protectable expression. Unfortunately, the line separating an expression and an idea is often problematic, and can only be determined on an *ad hoc* basis. In Judge Learned Hand’s words, “[n]obody has ever been able to fix that boundary, and nobody ever can.”<sup>6</sup>

To prevail in a copyright infringement action, the plaintiff must show that it owned the copyright in question and that the defendant copied expression protected by that copyright. The plaintiff may prove copying inferentially by demonstrating that the defendant had access to the copyrighted work and that the allegedly infringing material is substantially similar to it. The seminal case of *Arnstein v. Porter* established a bifurcated test for determining substantial similarity.<sup>7</sup> First, with the aid of expert testimony, the fact-finder decides whether the alleged infringer copied from the copyrighted work. This is known as the “extrinsic” test of substantial similarity. An affirmative answer triggers the second part of the test, called the “lay observer” or “intrinsic” test of substantial similarity. Here, without the aid of expert testimony and from the perspective of the lay observer, the fact-finder decides whether the copying was an illicit or unlawful appropriation; *i.e.*, whether the alleged infringer appropriated protected expression as opposed to an unprotected idea.

Two other points are important to keep in mind. The first is that computer programs are “literary works” within the meaning of the Copyright Act,<sup>8</sup> and that such works may be infringed by nonliteral copying if the result is substantially similar. The sec-

---

(b) 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. *Id.*

<sup>5</sup> *Atari, Inc. v. North American Phillips Consumer Elecs. Corp.*, 672 F.2d 607, 616 (7th Cir. 1982).

<sup>6</sup> *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930) (citations omitted).

<sup>7</sup> 154 F.2d 464, 468-69 (2d Cir. 1946).

<sup>8</sup> 17 U.S.C. § 101 (1988). “‘Literary works’ are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tape, disks, or cards, in which they are embodied.” *Id.*

ond is that the scope of copyright protection will necessarily be narrower with respect to utilitarian or fact-based works,<sup>9</sup> at least in part because of the idea-expression dichotomy described above.<sup>10</sup> Accordingly, the courts are confronted with a delicate task in applying these concepts to cases involving computer programs. On one hand, if courts protect too narrowly, there is a danger that the incentive to invest in new software products will be destroyed. Conversely, if they protect too broadly, there is a danger that basic principles of program engineering and interface design will be appropriated by software pioneers to the detriment of the public.<sup>11</sup>

The issue goes to some very basic and fundamental questions. How far do we protect and on what basis? The trend had been to give broader protection—and hence to limit the public use of copyrighted material. This was represented by the Third Circuit case of *Whelan Associates*,<sup>12</sup> and was followed by *Lotus Development*,<sup>13</sup> in the District Court of Massachusetts in 1990. One of our panelists, Henry Gutman, along with [Professor] John Beckerman [of Benjamin N. Cardozo School of Law], represented the prevailing party in that case.

Recently, the Second Circuit opinion in *Computer Associates International, Inc. v. Altai, Inc.*,<sup>14</sup> written by panelist Judge John M. Walker, Jr., rejected the *Whelan* approach. Another panelist, Susan G. Braden, represented Altai, Inc., the prevailing party in that case. The *Altai* court, noting academic criticism of the *Whelan* idea-expression formulation, rejected it as “descriptively inadequate.”<sup>15</sup> The court then proposed its own three-part substantial similarity test for computer program structure.

---

<sup>9</sup> *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 563 (1985). “Copying a news broadcast may have a stronger claim to fair use than copying a motion picture.” *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 455 n.40 (1984). “[T]he scope of fair use is greater with respect to factual than non-factual works.” *New Era Publications Int’l v. Carol Publishing Group*, 904 F.2d 152, 157 (2d Cir. 1990). See *Basic Books, Inc. v. Kinko’s Graphics Corp.*, 758 F. Supp. 1522, 1532-33 (S.D.N.Y. 1991) (citing cases).

<sup>10</sup> See *supra* notes 5-9 and accompanying text.

<sup>11</sup> See Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1047-48 (1989).

Drawing the line too liberally in favor of copyright protection would bestow strong monopolies over specific applications upon the first to write programs performing those applications and would thereby inhibit other creators from developing improved products. Drawing the line too conservatively would allow programmer’s efforts to be copied easily, thus discouraging the creation of all but modest incremental increases. *Id.*

<sup>12</sup> *Whelan Assoc’s v. Jaslow Dental Lab.*, 797 F.2d 1222 (3d Cir. 1986).

<sup>13</sup> *Lotus Dev. Corp. v. Paperback Software Int’l*, 740 F. Supp. 37 (D. Mass. 1990).

<sup>14</sup> 982 F.2d 693 (2d Cir. 1992).

<sup>15</sup> *Id.* at 705.

Although the *Altai* approach narrows the scope of copyright protection with regard to the program's nonliteral elements, it nonetheless accepts *Whelan's* conclusion that copyright may protect such elements. Thus, the vehemence of some of *Altai's* critics may be misplaced. [Dean Macchiarola concluded by welcoming the panelists and audience.]

*Professor Marci A. Hamilton*

[After several introductory remarks, Professor Hamilton continued.]

As one of our panelists, Jessica Litman, has cogently pointed out, our copyright law seems to be continually befuddled by developing technology. The quintessential example in the 80s and now the 90s is computer software. Congress has explicitly placed computer software under the aegis of the Copyright Act but it has left it to the courts to figure out how. As today's panel will ably demonstrate, this is quite a dilemma. As the courts have faced this dilemma they have been forced to address two different and difficult questions.

First, what is a computer program? This turns out to be something of a metaphysical question. Second, which element or aspect deserves protection? So far, no two Circuit Court opinions have mapped each other on both of these issues.

There is a third issue, however, that I would hope to introduce into the discussion today which I believe provides a key to understanding some of the differences and approaches in this area, and that is the role of the computer programmer as an artist. Should she be considered the equivalent of the imaginative author or artist of say a play or a poem, therefore, deserving the full extent of copyright protection; or is she more of a technician deserving a lower level of protection?

If one reads the cases carefully, one finds different depictions of the individual sitting there writing the program, and those differences may help answer the question why they would place protection at a higher or lower level.

To sum up, our symposium participants today will address mainly three questions. They are fundamental but they are also metaphysical.

1. What is a program?
2. What elements will be protected, and how do we know what those elements are?
3. What sort of author is the programmer?

Two cases lead the inquiry at this point.

One is *Whelan*, written in the Third Circuit by Judge Becker. The other one is *Altai*, written by our panelist, Judge Walker. As a former law clerk to Judge Becker, I have a hard time believing that *Whelan* could be wrong, but I promise to have an open mind.

[Professor Hamilton proceeded to introduce the panel.]

*Honorable John M. Walker, Jr.*

[After brief welcoming remarks, Judge Walker continued.] [W]e're going to be discussing an area that is still emerging; and we're not entirely sure of the final destination.

It's not unlike a story once told about Oliver Wendell Holmes in his later years. The Justice was on a train from Washington to Boston and the conductor came by asking for his ticket. Holmes fumbled in the pockets of his suit and then searched through the pockets of his overcoat, all the while bearing a nonplused look on his face. The conductor said, "Oh, Mr. Justice, don't worry about it at all, we'll contact you tomorrow. I'm sure you'll be able to find your ticket." Holmes replied: "No, no, young man, you don't understand, it's not that I'm missing my ticket, I don't know where I'm going." Well, in the same sense, I think, we're in an area of law that is being forced by technology to adapt, and we're not entirely sure what the ultimate destination is.

We have been invited by our hosts to address two questions which are fundamental to the continuing debate over the scope of copyright protection for computer programs. The first asks: "To what extent should copyright protect the nonliteral elements of computer software?" The second outlines an even more basic inquiry, that being, "whether copyright is the appropriate body of law in which to protect computer software?"

These questions seek normative rather than descriptive responses. They do not simply ask what the present state of the law is, but rather what it should be. In effect, they posit a blank slate and invite us to draft, *de novo*, a rational and comprehensive system of intellectual property protection for computer programs.

In meeting this challenge, my co-participants enjoy a significant professional advantage. As legal commentators, their function is to keep a vigilant eye on the law's development. Their jurisdiction is limited only by the power of their arguments. In addressing contemporary problems, they are free to abandon the status quo, and, if necessary, press for entirely new strategies.

As a judge, however, my role as a critic is more narrowly circumscribed. It is properly bounded by certain institutional constraints, such as respect for statute and precedent, and thus focuses on incremental innovation rather than wholesale re-evaluation. Judges spend the vast majority of their time applying the law in its "present" state. As a result, we are generally more comfortable with our descriptive skills than our normative ones.

For that reason, I shall concentrate my remarks on what has recently become the prevailing method of determining the scope of copyright protection for nonliteral components of computer software. My comments may be helpful to this discussion for the simple reason that, before commentators can argue constructively about what the law *should be*, it is wise to have a clear perception of what the law *actually is*. At the close of my remarks, I shall also comment briefly upon the nature of software protection under two separate doctrines: patent and trade secret.

But first copyright. Copyright subsists in particularized expression alone; it does not extend to the underlying idea or process which is expressed.<sup>16</sup> The debate over copyright protection for computer software centers on this age-old distinction. In *Computer Associates Int'l, Inc. v. Altai, Inc.*,<sup>17</sup> my colleagues on the Second Circuit and I set forth a three-step approach for analyzing this dichotomy in the context of computer programs. The primary focus of *Computer Associates* concerned the scope of copyright protection for program components that are not reduced to written code.

For the purposes of the Copyright Act, Congress has determined that computer programs are literary works.<sup>18</sup> The Second Circuit has long acknowledged that any meaningful protection of traditional literary property must entail, to some degree, protection beyond a work's strictly textual form.<sup>19</sup> Thus, to be consistent with decisions that afford copyright protection to certain literary themes, plot sequences and character types, our decision in *Computer Associates* recognizes that protection may also extend beyond program code to the analogous nonliteral elements that

---

<sup>16</sup> *Mazer v. Stein*, 347 U.S. 201, 217 (1954); *Baker v. Selden*, 101 U.S. 99, 103 (1879); 17 U.S.C. § 102(b) (Supp. 1993).

<sup>17</sup> 982 F.2d 693 (2d Cir. 1992).

<sup>18</sup> H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5667 ("to the extent that they incorporate authorship in the programmer's expression of original ideas" *id.*) [hereinafter House Report]. The definition of "literary work" is codified at 17 U.S.C. § 101 (1988).

<sup>19</sup> *Nichols v. Universal Pictures Co.*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931).

may be found in computer software.<sup>20</sup> Broadly speaking, these analogues are collectively known as program structure, and include modules, macros, parameter lists, and flow charts.<sup>21</sup>

Program structure gets no special treatment under the Copyright Act. It enjoys copyright protection only to the extent that it embodies expression rather than idea or process.<sup>22</sup> Furthermore, computer programs are highly utilitarian works, that is, they accomplish tasks. Thus, the goal of the *Computer Associates* panel was to find a legally accurate method for identifying utilitarian expression that is, in fact, protectable under well established rules of copyright law.

In so doing, our goal was to improve upon the "one program/one idea" rule adopted by the Third Circuit in the *Whelan* case.<sup>23</sup> *Whelan* held 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.*"<sup>24</sup> Considering, however, that a computer program is normally comprised of numerous "mini-programs," each with its own particular idea and function, *Whelan's* definition of program expression is over-inclusive.<sup>25</sup>

Learned Hand observed that, "[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."<sup>26</sup> At some point along this series of abstractions, Judge Hand reasoned that these patterns "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."<sup>27</sup> In the context of computer software, Judge Hand's test is superior to the *Whelan* rule because it recognizes that a single work may contain many ideas and expressions.<sup>28</sup>

While the abstractions test affords a workable template with which to begin designing a copyright infringement analysis for computer programs, it is only a first step. Taken alone, it does not sufficiently account for the utilitarian nature of such works. In other words, by simply refracting the program's conceptual

<sup>20</sup> 982 F.2d at 702.

<sup>21</sup> *Id.* at 697-98.

<sup>22</sup> *Id.* at 703-06 (citing House Report, *supra* note 18, at 5667, 5670).

<sup>23</sup> *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

<sup>24</sup> *Id.* at 1236.

<sup>25</sup> *Computer Assocs.*, 982 F.2d at 705.

<sup>26</sup> See *Nichols*, 45 F.2d at 121.

<sup>27</sup> *Id.* (relying on *Holmes v. Hurst*, 174 U.S. 82, 86 (1899)).

<sup>28</sup> See *Computer Assocs.*, 982 F.2d at 706-07.



spectrum,<sup>29</sup> the abstractions test may not reflect certain external constraints, which are dictated by the program's task. This is not surprising, though, since the test was first conceived in the context of novels and plays.

In order to allow for the change of medium, the panel utilized a "filtration" analysis as the second step in our three-part test.

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 . . . .<sup>30</sup> By undertaking this examination, a court can isolate those aspects of a program's structure that traditionally have been denied protection under the longstanding copyright doctrines of merger, *scenes a faire*, and public domain.<sup>31</sup>

Of course, this process also results in the identification of program structure that warrants copyright protection. By comparing this protectable expression to the structure of an allegedly infringing work, a court can determine whether infringement has, in fact, occurred.

This is the *Computer Associates* Abstraction-Filtration-Comparison test. While it is too early to tell how this method will ultimately fair in the judicial marketplace, the Federal and Ninth Circuits apparently have already embraced it.<sup>32</sup> Furthermore, the opinion does not purport to be the last word on the subject. Indeed, it explicitly acknowledges that the case law in this area needs to develop.<sup>33</sup> However, it also presupposes that a textually accurate reading of the decision will serve as the foundation for that development.

In this regard, I would like to address briefly two points of criticism, which I believe stem, in large part, from imprecise readings of the opinion. Thus far, perhaps the most unfavorable critique of *Computer Associates* has been offered in an article co-authored by Mr. Clapes.<sup>34</sup> Among other things, Mr. Clapes ar-

---

<sup>29</sup> See *id.*

<sup>30</sup> *Id.*

<sup>31</sup> *Id.* at 706-10.

<sup>32</sup> See *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839-840 (Fed. Cir. 1992); *Sega Enters., Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1525 (9th Cir. 1992, as amended 1993).

<sup>33</sup> *Computer Assocs.*, 982 F.2d at 712.

<sup>34</sup> See Anthony L. Clapes & Jennifer M. Daniels, *Revenge of the Luddites: A Closer Look at*

gues that by filtering the plaintiff's program before comparing it to the defendant's, our test "completely discarded the notion of comparing the original and accused works in their entirety, thereby trashing the principle that a selection, arrangement and organization of unprotected elements may itself be protected."<sup>35</sup> Mr. Clapes derives limited judicial support for his position from *Gates Rubber Co. v. Bando American, Inc.*,<sup>36</sup> a recent software copyright decision issued by the federal district court of Colorado prior to our decision in *Computer Associates*.

If our decision in *Computer Associates* had eschewed the principle that the selection and organization of non-protectable material may itself be the subject of copyright, I would agree with Mr. Clapes that it was in grave error. However, it did not. I say it did not simply because the opinion explicitly states that "[t]he functions of the modules in a program together with each module's relationship to other modules constitute the "structure" of the program."<sup>37</sup> The decision further states that "a program's structure includes its nonliteral components such as general flow charts as well as the more specific organization of *inter-modular relationships*, parameter lists, and macros."<sup>38</sup>

Thus, the opinion recognizes that program structure includes aspects of "selection, arrangement and organization." Contrary to Mr. Clapes' contention, the filtration of a program's structure does not preclude the possibility that the organization of certain unprotectable modules may itself be protectable. Rather, the filtration step, in part, concentrates the court's investigation upon whether or not a particular facet of the program's organization warrants protection. To the extent that a program's inter-modular relationships and general flow constitute expression, it will survive being filtered and contribute to the final comparative aspect of the infringement analysis.

At least one Court has recognized that the filtration analysis

*Computer Associates v. Altai*, 9 THE COMPUTER LAWYER, Nov. 1992, at 11 [hereinafter *Luddites*].

<sup>35</sup> *Id.* at 13 (endnote omitted).

<sup>36</sup> 798 F. Supp. 1499, 1516 (D. Colo. 1992). In fact, though, the *Gates* court did employ the filtration analysis outlined by the district court in *Computer Associates*, but only after it compared the two programs at issue in their entirety. See *id.* at 1516-20 ("[T]he application of the abstractions test, not instead of, but in addition to the two-step test [of *Whelan*], serves as a guard against unprotected elements being considered in the legal conclusion of whether there is infringement . . . ." *Id.* at 1516).

<sup>37</sup> *Computer Assocs.*, 982 F.2d at 698 (emphasis added; citation omitted).

<sup>38</sup> *Id.* at 702 (emphasis added) (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, 871 (1990)).

outlined in *Computer Associates* does not preclude affording protection to the organization of program elements. In *Atari Games Corp. v. Nintendo of America, Inc.*, the Federal Circuit applied our three-step infringement test to a computer program that was designed to prevent a video game system from accepting unauthorized game cartridges.<sup>39</sup> After filtering the plaintiff's program in a manner consistent with the *Computer Associates* approach, the Federal Circuit concluded that, "[a]t a minimum, [plaintiff] may protect under copyright the unique and creative arrangement of instructions in the [program at issue]."<sup>40</sup>

Another observer has criticized the analysis developed in *Computer Associates* for its "mechanical importation" of the merger and *scenes a faire* doctrines into the software arena.<sup>41</sup> The writer contends that "the liberal use of the merger and *scenes a faire* doctrines in the computer programming context is problematic because a programmer, unlike a novelist or a playwright, starts out with a finite number of creative choices."<sup>42</sup> The argument continues that "[b]ecause the concept of 'functionality' can apply to many, if not all, program elements, merger theoretically may prevent a programmer from securing a copyright for any of his significant efforts."<sup>43</sup>

In response, it is fair to note that the decision did not mechanically import these doctrines into the software context. Before applying the doctrines of merger and *scenes a faire* to the program at hand, the opinion evaluated the doctrines in relation to the unique issues surrounding computer software.<sup>44</sup> Moreover, at several points in the opinion we stressed that findings of program infringement are highly fact specific, and that each program must be analyzed on a case-by-case basis.<sup>45</sup>

Furthermore, the fact that a programmer may start out with only a "finite number of creative choices," or the accomplishment of a particular task through programming may entail "significant efforts," is not a sound justification for extending copyright protection beyond its statutory limitations. A limitation upon the number of expressive means by which to communi-

---

<sup>39</sup> 975 F.2d 832 (Fed. Cir. 1992).

<sup>40</sup> *Id.* at 840.

<sup>41</sup> *Copyright Law—Scope of Protection of Non-Literal Elements of Computer Programs—Second Circuit Applies an "Abstraction-Filtration-Comparison" Test*, 106 HARV. L. REV. 510, 513 (1992)[hereinafter *Copyright Law*].

<sup>42</sup> *Id.* at 514 (footnote omitted).

<sup>43</sup> *Id.* (footnote omitted).

<sup>44</sup> *Computer Assocs.*, 982 F.2d at 709-10.

<sup>45</sup> *See id.* at 710, 715.

cate an idea has traditionally militated in favor of relatively "thin" copyright protection in such works.<sup>46</sup> And "significant efforts" standing alone do not merit copyright protection.<sup>47</sup> While genius is born of both inspiration and perspiration, it is protected by *copyright* if, and only if, it constitutes expression.<sup>48</sup> Accepting these premises does not necessarily lead to the conclusion that long established copyright doctrines, such as merger and *scenes a faire*, are inapplicable to computer programs. Rather, these premises suggest that copyright may be the wrong vehicle for affording comprehensive software protection.

This, of course, brings me to the second question raised today: "Is copyright the appropriate body of law in which to protect computer software?" Some critics argue that the answer to this question entails economic policy determinations that are best left to Congress.<sup>49</sup> I agree. However, I part company with them when it comes to the rationale behind our mutual conclusion.

Simply put, they contend that we can force-fit the precepts of copyright law into the software niche, and thereby afford exhaustive protection. The *Computer Associates* decision, on the other hand, stands for the proposition that doctrines of intellectual property should not be distorted in order to accommodate hybrid works.<sup>50</sup>

At the risk of being called a Luddite by some,<sup>51</sup> I want to confront a mischaracterization that is at the core of our disagreement and, I believe, has done much to confuse this debate. Computer programs are not "just like" traditional literary works or works of art. While some commentators have analogized computer software to modernistic collages, whose collective arrangements alone gives rise to protectable expression,<sup>52</sup> the analogy is incomplete. To quote from Judge Keeton's most recent opinion in the *Lotus* case,<sup>53</sup> "computer programs, whatever their formal classification, like pictorial, graphic, and sculptural works, are useful articles." Unlike purely aesthetic works, whose "form is their essence," and whose "final end" is "the production of plea-

---

<sup>46</sup> See *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 111 S. Ct. 1282, 1289-91 (1991).

<sup>47</sup> *Id.* at 1294-95.

<sup>48</sup> *Computer Assocs.*, 982 F.2d at 711-12.

<sup>49</sup> *Luddites*, *supra* note 34, at 16.

<sup>50</sup> *Computer Assocs.*, 982 F.2d at 712.

<sup>51</sup> See generally *Luddites*, *supra* note 34.

<sup>52</sup> *Copyright Law*, *supra* note 41, at 515 n.44; see *Luddites*, *supra* note 34, at 12-13, 17 n. 34.

<sup>53</sup> *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203, 210 (D. Mass. 1992).

sure in their contemplation,"<sup>54</sup> computer programs actually accomplish tasks.<sup>55</sup>

I do not mean to suggest that utilitarian articles are, by definition, ineligible for copyright protection. That is not the law.<sup>56</sup> However, where functional elements of a work come into play, a more exacting copyright analysis is required so not to inadvertently remove ideas, procedures, processes, systems, and methods of operation from the public domain.<sup>57</sup> Therefore, in developing a satisfactory framework for the protection of computer software, it is important to be both honest and precise about exactly what we seek to protect.

Those who mourn the loss of *Whelan* may find some comfort in other intellectual property doctrines that are potentially applicable to program structure. Patent law, for example, is one possibility.

Unlike copyrights, patents are intended to protect ideas by granting inventors exclusive rights over their discoveries.<sup>58</sup> A patent holder need not establish copying in order to prove infringement. Rather, the test for liability is simply whether the defendant has made use of the plaintiff's patented technology or process.

By statute, the scope of patentable subject matter expressly includes "useful process[es],"<sup>59</sup> a category of inventions which would appear to encompass certain aspects of computer programs. However, until recently, courts and commentators questioned whether computer programs were, in fact, patentable.<sup>60</sup> This doubt arose from the established principle that patent protection should not extend to laws of nature, physical phenomena, and abstract ideas—a limitation that prevents inventors from obtaining patents on things which they did not invent.<sup>61</sup> While Einstein justly received the Nobel Prize for his pioneering discoveries in physics, it was equally just that he received no pat-

---

<sup>54</sup> *Baker v. Selden*, 101 U.S. 99, 103-04 (1879).

<sup>55</sup> *Sega Enters.*, 977 F.2d at 1524.

<sup>56</sup> *See Mazer v. Stein*, 347 U.S. at 217; *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1145 (2d Cir. 1987).

<sup>57</sup> *Cf. Brandir*, 834 F.2d at 1145 (holding that conceptual separability exists when "design elements can be identified as reflecting the designer's artistic judgement exercised independently of functional influences" *id.*).

<sup>58</sup> *See* 35 U.S.C. § 101 (1988).

<sup>59</sup> *Id.*

<sup>60</sup> *See generally* Randall M. Whitmeyer, *A Plea for Due Processes: Defining the Proper Scope of Patent Protection for Computer Software*, 85 Nw. U. L. REV. 1103, 1113-23 (1991) (discussing the history of courts' "varying conclusions regarding the ability and suitability of using patent laws to protect computer programs" *id.* at 1104 n.6).

<sup>61</sup> *See id.*

ent royalties for their subsequent application. Such scientific theories, mathematical computations, or algorithms,<sup>62</sup> have historically been identified as laws of nature. Since computer programs typically employ algorithms, some thought for a time that software could not be protected by patent.

However, recent case law makes clear that an invention's mere incorporation of mathematical procedures will not bar it from patent protection. Instead, "the mathematical procedures are considered in the context of the claimed invention as a whole."<sup>63</sup> Thus, if an invention applies a mathematical algorithm in an otherwise patentable process or apparatus, the invention may be patented. Applying this reasoning, the Federal Circuit has recently held that a medical testing device, which utilized mathematical calculations as part of a software package was potentially patentable.<sup>64</sup> Although computer programs that employ mathematical calculations may fall within the general scope of patent protection,<sup>65</sup> the extent to which programs may satisfy patent law's rigorous requirements of novelty and nonobviousness remains to be settled by future cases.

Another means of affording protection to software is the related state law doctrine of trade secrets. In *Computer Associates*, we recognized that trade secrets claims based upon the misappropriation of ideas, and not simply unauthorized copying, are not preempted by the Copyright Act.<sup>66</sup> Indeed, as we noted in the decision, "[p]recisely because trade secret doctrine protects the discovery of ideas, processes, and systems which are explicitly precluded from coverage under copyright law,"<sup>67</sup> it can serve as an important means of protecting the nonliteral components of software.

Trade secrets law protects just that—secrets. Thus, a plaintiff usually must prove that a defendant or its agent has misappropriated information that has been kept confidential by another, typically through violation of a contractual or fiduciary duty.<sup>68</sup> Trade secrets law affords no protection once information

<sup>62</sup> *Arrhythmia Research Technology, Inc. v. Corazonix Corp.*, 958 F.2d 1053, 1056 n.3 (Fed. Cir. 1992) (broadly defining an algorithm as "a procedure for solving a particular mathematical problem" *id.*).

<sup>63</sup> *Id.* at 1057 (relying on *Diamond v. Diehr*, 450 U.S. 175, 188 (1981)).

<sup>64</sup> *Arrhythmia Research*, 958 F.2d 1053.

<sup>65</sup> See 35 U.S.C. §§ 102, 103 (1988).

<sup>66</sup> 982 F.2d at 716-17.

<sup>67</sup> *Id.* at 717.

<sup>68</sup> See Restatement (First) of Torts § 757 (1939); see also *Balboa Ins. Co. v. Trans Global Equities*, 267 Cal. Rptr. 787, 794-96 (Cal. App. 3d Dist. 1990), *cert. denied, sub nom. Collateral Protection Ins. v. Balboa Ins. Co.*, 498 U.S. 940 (1990).

is public, and does not prevent reverse engineering of marketed computer programs to create functionally identical software packages. As a result, trade secrets law, like patent law, offers limited protection for software.

A patchwork system of existing copyright, patent and trade secrets law may still afford insufficient protection for computer programs. Whether that is the case or not is a legislative policy decision. The time seems ripe for Congress to take a second look at the question of whether computer programs should be afforded new statutory protections. There have been major changes in computer software and its role in the nation's commercial and cultural life since Congress last considered these issues.<sup>69</sup> Upon further examination, the Congress may well conclude that existing legal doctrines provide insufficient protection to computer programmers. When that time comes, you may rest assured that the courts will faithfully apply any new statutory protections that Congress may enact.

*Professor Jessica D. Litman*

Let me take the more global question first: Is copyright the appropriate body of law with which to protect computer software?

Ten years ago, I would have argued—indeed I did argue—that it was not. If we were beginning today with a completely clean slate on which to write software protection, I would still argue that copyright would be an unfortunate choice of a home for software protection law.

But we're not writing on a clean slate.

We have not only spent years adjusting copyright to software protection and software protection to copyright, we've also muscled all of our trading partners into adopting copyright protection for software. Some of them accepted the copyright paradigm only reluctantly and we would not be behaving as good world citizens if we suddenly announced, "Whoops, sorry, we were wrong about that; we're giving up on the copyright model."

So I think that whether copyright is the most appropriate vehicle for software protection or not, our decision to leave software protection within the copyright statute is a done deal.

What we've got to figure out now is how best to assimilate software protection to the copyright model.

---

<sup>69</sup> The last report commissioned by Congress was delivered in July, 1978. National Commission on New Technological Uses of Copyrighted Works, *Final Report* (1979).

Now, there are a bunch of reasons why I would have argued that, if you were designing software protection from scratch, copyright is not the model you would want to pick. The most fundamental of these reasons is the essential copyright distinction between expression, which can be protected, and “idea[s], procedure[s], process[es], system[s], method[s] of operation, concept[s], principle[s] or discover[ies],” which may not be. That’s a very difficult distinction to apply to computer programs. Programs are, after all, procedures and processes at their core. They are quintessential functional works. It’s their functionality that makes them valuable.

Now, I’ve heard some people argue, “Gee, copyright is for arty stuff like music, painting and novels, not for technology.” I am not making that argument, and I would not; indeed, I think that line is entirely illusory.

What I am suggesting is that as difficult as we have found it to separate the unprotectable idea, procedure or process from the expression in, say, a novel, it is infinitely more difficult when the work is a computer program, because the expressive part and the process part are pretty much the same. That leaves us with a perplexing problem.

*Baker v. Selden*<sup>70</sup> and section 102(b) of the copyright statute make clear that copyright can never protect the ideas, systems, processes and so forth, no matter how ingenious they are, how creative they are, how valuable they are. Section 102(b) is not merely this pesky little statutory section that we can ignore whenever it gets inconvenient. It’s a central axiom in the copyright scheme. So, one way or another, we need to come up with a workable way to apply it to computer programs.

What evidence we’ve got of what was going through the minds of the CONTU Commissioners when they recommended that Congress assimilate computer programs to literary works is that the Commissioners believed copyright would protect only the literal code and would indeed protect it against only verbatim copying, and that’s how we’d handle the *Baker v. Selden* problem.

But it doesn’t take very long to recognize that if the only thing copyright is going to protect is the literal code of programs against verbatim copying, and moreover, if the merger doctrine is there to privilege copying of any literal code that’s merged with the process or procedure or method of operation or system that it expresses, then the protection that’s going to give to computer

---

<sup>70</sup> 101 U.S. 99 (1879).



programs is hardly worth the trouble of writing "C-in-a-circle" on your disks.

Indeed, if you hold that copyright never protects any expression that's inseparable from the idea, process, procedure and so forth, it's difficult to see what in a computer program is left to be protected.

On the other hand, you don't want to create a computer program exception to section 102(b). You'd be giving out this incredibly powerful potential monopoly on how computers work and what they do, and while that monopoly will allow its beneficiaries to preserve their current market share for a longer period of time than would naturally be the case, by, in essence, allowing them to keep their competitors from developing competing products, it would be a very bad thing for the progress of science and software design. The result would be to skew innovation in favor of today's, and yesterday's, technology at the expense of tomorrow's. And while it's certainly in the interest of today's market leaders (and their counsel) to argue that that's precisely what we should do, we'd be really foolish to let all of those companies talk us into it.

So, what's been going on for the past dozen years is that courts and lawyers have been struggling to come up with ways of applying section 102(b) to software, and have come up with a variety of answers.

Now, we're all lawyers up here and we are all trying to claim that we are wearing the mantle of traditional copyright protection. None of us is suggesting anything that we will admit might be new. Indeed, we are all applying traditional copyright law to software (that's what we claim), as all the cases have (or so those courts have claimed). Me too. I am certainly (and, I would argue, correctly) claiming to be assessing the cases from the vantage point of traditional copyright analysis.

The most troubling line of cases, in my view, stretches from *Whelan*<sup>71</sup> to *Lotus Development*<sup>72</sup> because those cases fall into the trap of concluding that the most valuable or creative piece of a computer program must be the protectable expression part, and the idea, system, process or procedure portion must be the part that isn't creative or valuable.

The temptation to look at things that way, to define things

---

<sup>71</sup> *Whelan v. Jaslow*, 797 F.2d 1222 (3d Cir. 1986).

<sup>72</sup> *Lotus Development v. Paperback Software*, 740 F. Supp. 37 (D. Mass. 1990).

the way Judge Becker did in *Whelan* and Judge Keaton did in *Lotus*, is understandable.

Professor Hamilton, when she talked about what we were going to talk about, asked us whether or not the computer programmer is engaging in very creative expression and therefore deserves full copyright protection, or is a mere technician and therefore deserves less.

We all have an instinct to say that copyright ought to protect the most valuable part, the most creative part, the most difficult or expensive part of a work. It appeals to our sense of justice. But that's not the way copyright works or ever has.

What was creative and valuable about Charles Selden's books on bookkeeping was his new bookkeeping system and the forms he had devised to use with it, not his prose. But the Supreme Court held that copyright could protect his prose but not his bookkeeping system or bookkeeping forms.

Copyrights protected maps and directories since the first American copyright statute. What's most valuable and difficult to gather for maps and directories is the factual information they convey. Next to coming up with and interpreting the data, designing the expression is almost trivial. But copyright leaves the difficult, creative, valuable, information gathering unprotected.

What's most valuable and creative about cookbooks is the recipes for creating new and delicious food things, not the text describing how yummy everything is going to be when you make it, or the decision to include three chocolate cakes and one lemon pie recipe in your dessert section, or even the instructions for how to beat eggs until they're thick and lemon-colored. Making up the recipes is the creative part and the valuable part, not figuring out what words to use to express them.

Notwithstanding a wonderful Note written by a graduate of this institution a couple of years back,<sup>73</sup> the copyright on the cookbook is limited to the words and to the selection and arrangement of the recipes. It simply doesn't cover the recipes themselves.

It may seem unfair but it's fundamental. That's how copyright is designed to work. The purpose of copyright is not to reward authors who deserve it, but, instead, to encourage authorship. Protecting the market share of today's industry leaders is not necessarily the same thing as creating a legal environment in

---

<sup>73</sup> Malla Pollack, Note, *Intellectual Property Protection for the Creative Chef, or How to Copyright a Cake: A Modest Proposal*, 12 CARDOZO L. REV. 1477 (1991).

which tomorrow's industry leaders can create protectable works, and copyright is supposed to do the second, even though, of course, today's industry leaders would much prefer that it do the first, and will do whatever they can to use copyright to that end.

Section 102(b)'s exclusion of copyright protection for ideas, processes, and so forth isn't there because ideas, systems and processes are less valuable. Often the ideas, systems, processes, and so forth are the most valuable, most creative parts of copyrighted works. That was so with Charles Selden's bookkeeping system; it's the case with maps, with cookbooks, and with most computer programs. Section 102(b) precludes protection for ideas, procedures, processes, systems, methods of operation, concepts, principles or discoveries so the rest of us can use them to make new and different copyrighted works, and thus contribute to the progress of science and the useful arts.<sup>74</sup>

So, the *Whelan* and *Lotus* approaches of assimilating expressive to creative or difficult or valuable really missed the point.

Now, if you've been listening to me so far you're not going to be surprised to hear that I think that one of the best things about Judge Walker's opinion in *Computer Associates v. Altai*<sup>75</sup> is that it (finally) comes to grips with the need to attack the section 102 problem in some meaningful way.

The *Altai* test isn't perfect, because courts will need the assistance of experts to identify the aspects of computer programs that are dictated by considerations of efficiency, or required by external factors, or taken from the public domain. That's going to be expensive, but in a dozen years no one has come up with a less cumbersome, less expensive way to pay real attention to section 102(b) in the context of software copyright.

This brings me to the second question, which is whether copyright should protect the nonliteral elements of computer software. That's not really the right question; the right question is the question the court asked in *Altai*. Under what circumstances, and subject to what limitations, should copyright protection of computer software extend both to code and noncode elements of programs?

I don't think many people are arguing any longer that the code/noncode line is the best line to draw. What we are trying to

---

<sup>74</sup> Or, in the words of Justice O'Connor: "This result is neither unfair nor unfortunate. It is the means by which copyright advances the progress of science and art." *Feist Publications v. Rural Tel. Service Co.*, 111 S. Ct. 1282, 1290 (1991).

<sup>75</sup> *Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992).

identify are the limitations and the circumstances under which we protect both code elements of programs and noncode elements.

Now, what do I mean by limitations and circumstances? An example of a limitation on copyright protection of a program's literal code is the fair use exception recognized in the *Atari*<sup>76</sup> case and the *Sega*<sup>77</sup> case for reverse engineering and decompilation of programs. That's a limitation on protection of the literal code.

The filtration of elements dictated by efficiency called for in the *Altai* opinion is a limitation on protection of both code and noncode elements. Copyright protection of literary works other than computer programs has always been subject to exceptions, conditions, and limitations. Some of those exceptions, like idea/expression merger, or the nonprotection of facts or the *scenes à faire* doctrine, have names and well-articulated rationales. Others are implicit in our analysis, either as part of the overarching idea/expression distinction or tacitly included in our concept of substantial similarity.

Now, Tony Clapes has argued elsewhere<sup>78</sup> that the gist of these limitations comes down to whether something is essential, which comes down to the question, "Is this the only way to say this or are there other ways?" And, certainly, there is language in the *Apple* case,<sup>79</sup> and in *Whelan*, that would support that kind of argument. But it's a misleading picture.

Again, consider Charles Selden's bookkeeping system. The forms the Supreme Court held unprotectable were essential only if one wanted to keep books the same way Charles Selden kept books. There are lots of other ways to keep books. Or, to steal one of my favorite examples of Tony's,<sup>80</sup> consider four cookbooks. Each cookbook has a recipe for chocolate truffles and the recipes are completely different, the ingredients are completely different. No particular formulation is essential for making chocolate truffles, yet copyright stops at the words and the ordering and does not extend to the actual chocolate truffles recipes.

There are, in fact, a slew of limitations that courts traditionally apply to literary works. Most of them prohibit protection of aspects of the works that are not trivial, are not uncreative, are

<sup>76</sup> *Atari v. Nintendo*, 975 F.2d 832 (Fed. Cir. 1992).

<sup>77</sup> *Sega v. Accolade*, 977 F.2d 1510 (9th Cir. 1992, *as amended* 1993).

<sup>78</sup> Anthony Lawrence Clapes, *SOFTWARES: THE LEGAL BATTLES FOR CONTROL OF THE GLOBAL SOFTWARE INDUSTRY* 17-19, 28-32, 61-66, 301-02 (1993).

<sup>79</sup> *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983).

<sup>80</sup> Clapes, *supra* note 78, at 301 n.4.7.

not the only way to put something, but are aspects that other authors need to use in order that authorship be encouraged.

Now, courts developed and applied those limitations almost intuitively,<sup>81</sup> because by the time we had a copyright statute protecting traditional literary works, we had many generations of exposure to and sophistication in analyzing traditional literary works.

In comparison, we are terrible naifs when it comes to dissecting, analyzing, and understanding computer software. Our children will do better, but in the meantime we have the responsibility of coming up with exceptions, conditions, and limitations that will best promote vibrant innovation and competition in the software field. And from that viewpoint, I would argue that the questions asked by the *Altai* opinion are precisely the right questions.

As a first pass at what part of a computer program is a procedure, process, system or method of operation, it's useful to ask: What aspects are dictated by efficiency? What aspects are dictated by external factors?

The next questions we are going to find ourselves confronting are the questions about what sorts of other aspects of computer programs are things we need to keep in the public domain so that other software writers can use them.

*Susan G. Braden*

The *Altai* case is in a peculiar posture right now so I am not going to talk directly about the case. The trade secrets issues are back before Judge Pratt on remand and there is an issue about whether opposing counsel should have asked for certiorari last week before the Supreme Court as to the copyright findings. I am going to talk in a more generic way. Essentially, I want to make several unrelated comments about the copyright law and the case. An overview of my remarks could be entitled "Madison's Cautious View of Copyright and Monopoly Power, as reflected in the *Altai* decision."

First, I recently finished writing a brief for the Second Circuit in a fair use case. Because of that work, I spent a lot of time going back through Madison's papers. It was very helpful to me to do that, because I did not focus on the historical aspect of the copyright law when I was working on the *Altai* case. It is likely

---

<sup>81</sup> For a notorious example, see *Stowe v. Thomas*, 23 F. Cas. 201 (C.C. Pa. 1853).

that the students here already understand that history better than those of us in practice.

As you know, Madison was the Founding Father who spent the most time thinking and writing about copyright. Madison was highly suspicious of monopolies, as were many of the Founders, for very good reasons. With the development of the printing press the printed word became a powerful influence in how governments were shaped. Those in authority utilized their control over the press to censor and monopolize ideas by allowing only certain works to be printed.

By the mid-16th century in England, the only books printed were those specifically authorized by the Crown. A cartel of stationer companies were established as the official publishers. They had the extraordinary power to search and seize unauthorized works and even to adjudicate actions against competitors.

In 1709, the British Parliament enacted the Statute of Anne, which conveyed a copyright over written works for 14 years plus one renewable term. When that time period lapsed, the established publishers began to initiate law suits against publishers outside London—who were viewed as the “pirates” of those days—to challenge what was in the public domain.

It was against this background that Madison and the other Founders considered the place of copyright in the Constitution. In forming this radical new form of government, democracy, they questioned to what extent should a monopoly over written works be conveyed by way of copyright?

To better understand the Framers’ state of mind, I would suggest looking at 37 *The Quarterly Journal of the Library of Congress* 2 (Spring 1980) and Volumes 13 and 16 of the *Harvard Journal of Law & Public Policy*, respectively the Summer 1990 and Winter 1993 editions, which discuss the nature of copyright and reflect in general on the Federalist papers. In reading these references, I was reminded that when someone died in the 1700s an inventory of their property included: how much land one owned, how many slaves one had, how much silver and how many books were in one’s possession. Books were so valuable because of the high cost of printing. In fact, I learned that Madison was forced to sell his book collection twice when he was working on the Constitution because he ran out of money and it was one of the few ways that he could raise funds. At that time, Madison was 27 or 28 years old, which is the age of many of the students in the audience. While Madison was considering what types of powers should be put in the Constitution, in his “spare time,” he also

composed an inventory of 1,300 journals, many of which were copyrighted by the Crown. Madison felt these works were important to get into the hands of the public. Unless they understand the philosophical and economic learning that had come into being, how could democracy ever work? Only those of wealth and privilege at that time, however, were literate and even had access to these works. It wasn't until Madison became president that he had the opportunity to purchase books from Thomas Jefferson's estate, which became the core collection of today's Library of Congress.

Madison, as the other Founders, was very suspicious about copyright and whether it would inhibit the type of public debate that they wanted to have in the new world—in this new social experiment, called democracy.

My second point is the Constitution speaks of copyright as a defined and limited monopoly right. Indeed, the word "limited" was specifically part of the language of Article I, Section 8. Copyright was not considered to be a natural or inherent right, but rather one granted by government in order to promote the public interest. Copyright was designed as an economic tradeoff between private incentives and social benefits. The government has the power to convey the monopoly. The monopoly is to compensate for market imperfection.

Today's question is do we have an imperfect market in computer software? Perhaps at one point we did, when the manufacturing of hardware began. Since the first computer manufacturers were the only one that knew how the machines worked, perhaps they had an undue influence over the type of software that was initially developed. Certainly, however, this is no longer the case.

With these historical observations in mind, let me briefly turn to the *Altai* opinion, written by Judge Walker of the Second Circuit, and make a few more practical observations. The courts have been left to decide which part of these computer programs are the written expression of ideas and which part merely describe the way a computer functions. The Second Circuit's analysis essentially is a set of instructions to the district courts. For those of you who have not had an opportunity of trying a lawsuit recently, by and large, the district court docket consists of criminal cases. The amount of time that can be devoted to any civil matter is limited, particularly one of a complex nature, such as the *Altai* case.

What Judge Walker has done, I believe, is attempt to give

the district courts an analytical model that can help guide them not only in trying the lawsuit but in determining how to get a handle around the evidence of the parties throughout them. In that sense, I think it is extraordinarily helpful. The second practical point I want to make is you can look at the abstraction-filtration-comparison test, and I'm going to tell you the action is going to be in filtration obviously; what goes in, what stays out. There is an element that is not in the filtration analysis by the court, but somehow is, I think, perhaps inherent in the efficient use of the test, and perhaps inherent in the idea/expression portion of it as well, and that is this: all of these cases that are being litigated—and it's great to talk about these things on a high intellectual plane or moral basis, but—somebody generally is trying to protect a market that they either have legitimately or don't. Moreover, they have alleged that someone has taken and used something inappropriately, and they may have or they may not have. I suggest to you that the equity powers of the district court are very important in how these cases are going to be determined, and I think you'll find this to be true by looking at how they view these filtration elements.

"Cat on a Hot Tin Roof" has a wonderful scene in it where Burl Ives, who plays Big Daddy, says, "There is a powerful smell of mendacity in this room." I suggest to you that in all of these cases, a powerful aroma somehow will rise to the surface during the trial, and I suggest to you that that aroma of mendacity on one side or another will be what dictates how some of this filtration perhaps may be done.

I do not believe — Judge Walker was the trial judge and I'm sure he can attest—that there is anything more magnificent or awesome than to watch a judge exercise his equitable authority. The trial judges in this country have more power than any member of the Supreme Court ever has or ever might hope to have, and those judges are very skilled in using that equitable power and in one sense attempting to convey that in their decisions to the Court of Appeals. And the Court of Appeals knows that too. My sense is that when they receive these cases they can tell when that "fairy dust" has been sprinkled around an opinion.

Another practical consideration is the three-part test that Judge Walker has suggested I believe really compels a thoughtful district court to consider utilizing an expert for him or herself. This is a difficult thing because judges don't like to give up authority, and I am not suggesting in any way that the district court should bring in someone else to make factual findings for them



as a special master. To the contrary, as an advocate I don't want someone else doing that, I want the judge making that decision. I do think, however, that the experts have an important role to play in terms of translating for the judges and explaining to them what it is that the parties and the individual advocate's experts are actually saying to the court. I think that it's really almost — the judge didn't require this in the decision, but I think it's almost— required by looking at the types of decisions that are going to have to be made in this filtration area. With any amount of intelligence, this is a very difficult area. I think very few of the lawyers understand what's going on in these cases. I certainly didn't. I must tell you that I did not know a thing about computer software when I got involved in this case and I've got to tell you I think it was very helpful to me because I looked at the case very differently. I mentioned to you the equities. This case was brought the day my client was getting ready to merge with someone else, so I suggest now that perhaps the case has something to do with who should be or should not be in a market rather than whether or not the copyright laws have been violated. It's a little footnote that you don't see any place in the Court of Appeals opinion. Certainly it was a factor that was understood by the trial judge.

Finally, my last practical remark is to follow-up on Judge Walker's suggestion that we throw this to the Congress. Since I'm from Washington and have had some exposure to that process, I would suggest that's a terrific idea. I think they don't want it, number one, and number two, I think because of the way decisions are made, they're by and large — either you've got to have — is the party that has the stronger public interest or the largest pocketbook. And I think one of the reasons is there is not a large pocketbook for the public interest in this particular area, and it's one of the reasons I think those particularly in the judiciary committee have tried to work out problems in this particular area by consensus. Perhaps it's time for another CONTU, perhaps it's time for some thoughtfulness in terms of the types of guidelines that should be done, and perhaps that would be a useful thing for the industry to do. I hope this will give you something more to think about than just an analysis of the cases. The most interesting cases coming down the pike are all district court cases. There is *Gates Rubber*, *Consultech*, *Auto Skill*, *C-Max*, *Comprehensivetech*, and *Borland*. None of those cases I believe have gotten into the Court of Appeals yet.

But all of these cases have allowed nonliteral protection and

many of them have done it with variations on a theme in following the analysis of the Second Circuit more or less to different degrees. And I suggest to you that that resolve is reached probably by the factor that I left you with earlier, which is this mendacity power in the court when the facts are presented to the courts. We were asked to address whether "look and feel has crashed." I think the protection and scope of the "look cases" has narrowed. In "feel," I think they may well be expanding. I think it's too soon to know. But I think it is an exciting area to be in. I've enjoyed litigating in it, I've enjoyed learning about it, and I suggest it's an area perhaps where generalists really may have more to bring to the courts than the specialists because it's an area that is evolving and it's not static.

*Anthony L. Clapes*

There are moments when no matter how much speaking one does or how much writing one does, one is left speechless. I had such a moment not too long ago. Three or four weeks ago, I had to call Minneapolis to talk to a lawyer, a colleague of mine out there, and I didn't have the number with me, so I dialed the information operator. The operator got on the line and said, "Hello, this is Elvis, may I help you?"

I caught myself for a moment. It was an opportunity that cried out for some kind of a remark, but I couldn't come up with the proper response to that. Maybe, "How about a few bars of 'Blue Suede Shoes'," I don't know. While I was thinking, I could hear Elvis on the other end of the line thinking, "Oh, another joker; here we go again," and so finally I let the moment pass.

There's been a moment like that this morning. I never thought that I would be pleased to hear a distinguished member of the federal judiciary tell me that my interpretation of a case was based on careless or inadequate reading, but I certainly was pleased to hear Judge Walker say that to me this morning, because it corrected a view that I had had—that *Computer Associates* deprived, and indeed sought to deprive, computer programs of a form of protection that is generally available to copyrighted works.

IBM's new notebook computers. . . . [have] the processing, memory capacity, and disk capacity equivalent of a computer that would have taken up [an entire room] years ago. [It] costs at most a few thousand dollars where that earlier computer probably cost a few million dollars. [W]hat that march of technology means is

that, more so than ever before, the computer business is driven by software, not by hardware. [Personal and notebook computers] are being sold in the millions of units. The big main frame computers sold in, perhaps, the tens of thousands.

Software is very important to the continuing success of the computer industry, and is becoming an increasing part of the hardware suppliers' business. And I must say that for th[ese newer] computer[s], as for the big main frames of years ago, most of the software . . . is not, and never has been written by the hardware manufacturer. It's written by software houses and, by customers writing programs for their own use.

Now that software has overtaken hardware as the driving factor in the industry, the question of protection of software has become more and more important to industry participants, both those who only write software and those who both supply hardware and write software.

For a hardware supplier like IBM, the interesting question is, what software is available to run on these computers? The more diverse software there is, the more homes will be found, not just in [houses], but on business desks and elsewhere, in academe and so forth, for the hardware platforms.

The [impetus] that causes more software to be written—traditionally at least, there may be some other concept that we can come up with—but traditionally what causes software to be written to run on these computers is some kind of economic privilege or advantage that results from writing the software. If a customer writes software for herself or himself, it's because there is the hope that that software will bring some advantage that other people don't have. If the customer is an industrial [or] . . . commercial enterprise, the hope is that there will be some advantage in the work environment that will make the customer more competitive. If the customer is a software house, or someone who's writing software for a profit, then the hope is there will be a return on investment adequate to cover what was spent up-front in the development plus some kind of return that will help to fund future developments. The funding of future software developments, by and large, comes from profits. It doesn't come from venture capital. Increasingly, venture capital has tended to avoid the software business. Now, what is it that allows profits to be made? It is intellectual property protection. Software is almost pure intellectual property. You don't need a factory to bring it to the marketplace. Once it's written down, subject to being tested—it can be sold without being tested but testing always

helps—it's there, ready to be put on to diskettes or tapes or CD ROMs ("Compact Disk Read-Only Memory") and be distributed.

A competitor has no great obstacle to the copying of the software unless it's illegal. If it's not illegal to copy the software, then there is no business and there is no profit out of which to fund future developments. [Therefore, the] people who create the platforms on which the software runs are left with a non-expanding market.

With that little piece of industrial orientation out of the way, I would like to say that I don't think it's appropriate to talk about copyright as a monopoly at all. Copyright does not preclude independent development of similar or identical works, and one of the elements of monopoly is the power to preclude competition. Copyright law only precludes copying, and the other things that go along with copying, that are the subject of the copyright law: distribution, adaptation, translation, or creation of derivative works.

So I don't think that it's right to speak in terms of a copyright monopoly. It's a privilege that lasts for a limited although fairly lengthy period of time, and it is applicable, among other things, to literary works where literary is defined in a very broad sense, a sense which indeed encompasses computer programs.

Now, is it appropriate to protect software by copyright law? The Commission on New Technical Uses of Copyrighted Works ("CONTU") thought so; recommended to Congress that they do so; and Congress did so.

In the deliberations of CONTU, there was discussion of the fact that computer programs were functional. [I]n my view, nothing new has been added to the debate over whether computer programs are functional or not since then.

As to whether CONTU focused on the identical copying, the mechanical reproduction of computer works or not, I [can't say,] I wasn't there. One of the members of CONTU [, however,] was Mel Nimmer, the late, great copyright scholar[.] [H]e had a view on this question which is expressed in a declaration that he supplied in a lawsuit, which . . . has been reproduced in two law review articles, one of which I co-wrote. [H]is view is not to the effect that copyright as envisioned by CONTU for computer programs should be limited to the literal copying only, but rather that it should protect the traditional nonliteral elements that would be protectable, assuming that they were elements of expression, in other works.

One other aspect of copyright protection that's quite impor-

tant for the software industry is the fact that copyright is a world-wide regime put in place, and held [there], by international compact among nations. What that means is that the software industry can expect reasonably consistent copyright protection from country to country—not identical but reasonably consistent—and that allows a predictable business to be made out of writing software around the world.

Outside the [United] States, there are countries in which software copyright protection has been under attack for a long period of time. Governments in these countries look to the United States as a leader in this area because our law is better developed; we've been pursuing case law in the software protection area for much longer than most other [nations]. A decision like *Computer Associates* is seen in these countries as ratifying the notion that software ought to qualify only for weak protection by copyright. That's not good for the software industry.

The things that bother me about the *Computer Associates* decision are, number one, the notion that writing software is a highly constrained activity, and second, the filtration analysis.

I don't really have time to go through it, but in my latest book, [*Softwars*,] . . . I quote a number of people who are engaged in the business of writing software, as to whether it's an artistic or scientific process, and, as to whether it's a heavily constrained or heavily original process. The quotes are quite compelling.

People that write software think of it as a creative process, not as a highly constrained process. Where that notion came from in the *Computer Associates* opinion is a subject for another day I think; it's not correct. It's just flat incorrect. And yet the notion was, as the judge wrote in his opinion, vital to the decision.

What we see now in the *Atari* and *Sega* cases—the Ninth Circuit and Federal Circuit cases that Judge Walker mentioned, is two courts that have picked up on the theory that writing software is a highly constrained activity. They then used it to demolish another element of the copyright law—the fair use exception—by creating a special rule based on a really quite limited fair use analysis for computer programs, that allows for translation under some circumstances of the entirety of computer programs.

As to the filtration analysis, there isn't time to go into the details of the critique in my *Computer Lawyer* article.<sup>82</sup> It remains my view, though, that what the filtration analysis does in a way is

---

<sup>82</sup> See Clapes, *supra* note 34.

shift the burden of proof from the copyright defendant—whose job it normally is to bring forward defenses to demonstrated similarities—to the copyright plaintiff, who must in the Second Circuit prove at the outset that certain elements, or perhaps each element of his or her program is protected.

I want to leave you with a vision of the future in this area. I brought with me four CDs. The first one is what's called the *Oxford English Reference Library*. It's a computer program and a series of data files. The data files, if they were delivered to you in hard copy, would consist of the *Concise Oxford Dictionary of Current English*, the *Oxford Thesaurus*, a couple of quotations dictionaries, the *Complete Works of Shakespeare*, *Alice in Wonderland*, and the *Revised English Bible*.

When these things come up on the screen pursuant to your search, one might wonder whether one is looking at a user interface or a book, and whether it makes any difference under copyright law. More important though, what we have here is material delivered in digital form, 0s and 1s, on CD ROMs that run in a computer and cause the computer to produce results.

....

What these CDs demonstrate so eloquently is that the *Computer Associates'* world view is not only inaccurate, but obsolete, and will be rendered even more so as this kind of intermixing of media of expression becomes more and more prevalent, which it will in just a short number of years.

*Henry B. Gutman*

Since some of the matters I will talk about today are still in litigation, I should start with a disclaimer, making explicit what should already be obvious. In appearing here today I am not speaking on behalf of Lotus or any other client, and I am not speaking on behalf of my firm. The views I express are at best my own, and since I reserve the right to play devil's advocate, without saying so, some may not even be my own views.

So to the extent that anybody is here today with the hope of picking up an admission, please put down your pen and enjoy the presentation. There will be none.

Second, whatever his strengths or weaknesses as a reader, I would like to point out that Tony Clapes is a very skilled and distinguished writer in this field, and for those of you whose interest in this area is piqued by today's discussion, I would urge

you to read Tony's two books in this area.<sup>83</sup>

Now, the two questions posed today are fairly simple and straightforward. The first concerns the proper scope of copyright protection for nonliteral elements of computer programs, and the second whether copyright is in fact the best legal vehicle to provide that protection. I think the answers to these questions are also fairly simple and straightforward.

As to the first—that is, whether copyright properly protects the nonliteral elements of computer software—I think we are all in agreement that every case to consider the issue, from *Whelan* through and including *Altai*, has concluded that such protection is proper. As the *Altai* court put it, there was no reason not to join that long list of other courts that had concluded that when Congress deemed that computer programs are to be treated as literary works, that included providing protection for their nonliteral elements. I believe there is consensus on the principle. There is however, as you have probably already gathered, a fair amount of disagreement as to the price.

As to the second question, I frankly believe that copyright, however difficult it may sometimes be, is indeed the correct principal vehicle for protecting computer software. It is also quite clear that Congress intended it that way. There is no question that Congress meant for copyright law to protect new technology. If you look at the language of section 102(a) of the Copyright Act, it says: "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 a device."<sup>84</sup>

When Congress wrote that in 1976, it is quite clear—and would have been even without the post CONTU 1980 amendment—that it meant for new technologies such as computer programs to be protected by copyright. To the extent time permits, I will return to this subject, because I think Congress made the correct call.

Now, in addressing the question of defining the proper scope and degree of protection for nonliteral elements of computer programs, the issue can arise in two rather different factual contexts. The first context—that in which *Altai* arose—is one in

---

<sup>83</sup> ANTHONY L. CLAPES, *SOFTWARE, COPYRIGHT COMPETITION: THE LOOK AND FEEL OF THE LAW* (1989) and ANTHONY L. CLAPES, *SOFTWARES: THE LEGAL BATTLES FOR CONTROL OF THE GLOBAL SOFTWARE INDUSTRY* (1993).

<sup>84</sup> 17 U.S.C. § 102(a).

which the alleged infringement involves the internals of a program—that is, its source or object code. The issue arises when alleged copying is not verbatim, express duplicating of the code itself, but rather the copying of its whatever; since *Whelan* we have referred to it as its structure, sequence, and organization. Such things are comparable, in the literary context, to the plot of a novel or play.

The second context in which this issue arises, and this is the area where most of my efforts have been focused, is when you are dealing with the externals of the program, such as the user interface of the program, its screen displays, menus, and elements of the sort, rather than the literal code itself.

I would like to deal with both of these issues separately because I think they are distinct issues and some of the factors that come up are different depending upon which type of case is involved.

Obviously the leading decision in the field addressing the nonliteral protection of internals is *Altai*. I, too, was pleased and relieved to hear Judge Walker say that my friend Mr. Clapes was not, in this instance, as good or careful a reader as he might otherwise have been. Based upon her criticism of Judge Keeton's decisions in *Paperback* and *Borland*, I would suggest that Professor Litman should probably enroll in the same remedial reading course, because I think she missed the point as well.

The important point in *Altai*, I think, is to distinguish that which the court actually decided from the claims of those who have exaggerated and distorted its reach.

There is an army of people out there—a number of them in academic positions, a number of them employed in industry—who have a fundamental hostility to the notion that there should be effective copyright protection for computer software. In this day when we have all come to know what “FOBs,” or “Friends of Bill,” are, I would refer to these people as “FOCs,” or “Friends of Copying.”

To give you an example of the extraordinary hyperbole generated about the opinion, I learned that the Second Circuit decision in *Altai* had been handed down when a Wall Street Journal reporter called to tell me that he had just heard from Borland's counsel that we had just lost the *Lotus v. Borland* case.

Now, I was already familiar with *Altai*. I had certainly read the district court opinion. I had read Professor Davis's report. I had read all the briefs in the Court of Appeals, and I was familiar with the arguments that were made. So frankly, I did find it a



little difficult to understand how the ruling in that case—even if one ignored the fact that the First Circuit isn't the Second—could have possibly meant that we lost. But certainly that was the view of Borland's counsel, that's how it was widely reported in the press, and that's how it was argued by Borland to Judge Keeton. Fortunately, that was not how it came out. At least not so far.

But at a Copyright Society dinner, I heard Ms. Braden tell the group, in substance, that based upon her *Altai* victory, computer software copyright protection was dead. She said she was now advising her clients that it was not even worth the effort or the fee to file for a copyright registration; trade secret or patent protection, if you could get it, was the way to go.

As I read Judge Walker's *Altai* decision, it did not do any of those things. I would urge all of you to read the opinion with care, and to reject its more apocalyptic interpretation. I would also express the hope, before moving to a few specifics, that when lawyers for copyright infringers appear before the Second Circuit, having read and argued the opinion too broadly, Judge Walker's colleagues will be as diligent in correcting them concerning their reading as Judge Walker was with Mr. Clapes today.

On its facts, taken as the court found them, I think *Altai* may well have been properly decided on the copyright issue. If one looks at the three-part abstraction-filtration-comparison test, I think it has much in common with tests adopted by a lot of other courts, including the test that was used by Judge Keeton in the *Lotus v. Paperback* case and again in the *Lotus v. Borland* case. Its purpose is not unlike that of the infringement tests used in many other courts. I believe that if you focus on the fact that this is a substantial similarity test, even some of the elements of it that the FOCs might be most inclined to misuse do not stand for the propositions the FOCs assert.

For example, let's look at the question of efficiency. If *Altai* stood for the proposition, and I have seen it so argued, that efficient programs cannot be protected by copyright, and efficiently written code cannot be, then that would clearly be wrong as a matter of copyright law.

It would not just be wrong, it would be silly.

Imagine, for example, a like rule applied to more traditional literary works. "To be or not to be." "Call me Ishmael." It is hard to imagine more efficient prose than these lines. There is not a wasted word.

Obviously in more traditional literary media, writing efficiently is not something that causes an author to lose his other

copyright protection, and it does not in computer software either. There is no such principle under copyright law.

If, on the other hand, we remember that the principal purpose of this type of substantial similarity test is to compare similarities in two programs to determine whether they are of a type that would suggest to the fact finder that one was copied from the other, *Altai's* treatment of "efficiency" under this evidentiary rule made sense. Remember that the issue in *Altai* was whether the similarities upon which the plaintiff relied were sufficient to overcome the defendant's proof that the program had instead been copied at an efficient level.

Does the structural similarity give rise to an inference of copying?

If that is the focus, then quite obviously the fact that two programs were both written using an efficient programming technique would not suggest copying. It might suggest efficiency, or skill, but it would not necessarily suggest copying.

So, too, with compatibility. In *Altai*, one of the issues was the fact that both the allegedly infringing and the allegedly infringed programs were written to provide compatibility with a common operating system.

Now, when viewed under a substantial similarity test, that fact obviously might explain some level of similarity. Indeed, at a certain level (and in certain particulars), probably any application program written to work with a particular operating system will at various points in the program have to have certain characteristics in order to work at all. It is like a key fitting a lock. The fact that two programs are similar only to that extent does not, without more, prove that one was copied from the other.

In such a case, you might also argue that these details are not protected because they are not original, since the shape of the key was dictated by the operating system, an external factor. Dictated. Note my emphasis on the word "dictated." Being suggested, influenced, or vaguely related to functional or efficiency concerns does not give you a defense in a copyright case. Being *dictated* by such concerns—as in the elimination or virtual elimination of choice—may support a merger defense.

So, as I read the *Altai* decision, it is consistent with the preexisting framework of copyright protection under which computer programs are protected as literary works.

I have argued, and Judge Keeton has ruled, that there was no material inconsistency between *Altai* and the decisions en-

tered in *Paperback* and thus far in *Borland*. I happen to believe that that's right.

The key is to read *Altai* in a lawyerly fashion and to bear in mind the distinction we all learned between the holding of a case and its dictum. But there is great danger here because the FOCs are arguing very aggressively based upon *Altai*.

Before moving on, let me share a recent anecdote illustrating the potential danger.

I argued a case in the Second Circuit not long after *Altai* was decided. It was a piracy case in which we alleged that the defendants had physically copied the magnetic tape upon which the plaintiff's computer program was stored.

One of the questions from the panel was: "Why didn't the district court conduct an *Altai* analysis before granting an injunction?" I gulped, thinking I might be in real trouble, and explained that substantial similarity was not an issue below because this was a piracy case, and there was no question about the copying. It was a complete physical duplication of a tape, just like copying a disk or copying a video tape.

Indeed that precise issue was litigated before Judge Skinner in *Data General v. Grumman*<sup>85</sup> not too long ago. Judge Skinner ruled that you do not have to get into that type of analysis in a case where literal copying, as opposed to non-literal copying, is at issue.

But these cases point out the potential danger if *Altai* were carried too far or misapplied. We all, I think, have to be quite vigilant on this point.

Now, in the user interface context, I have just a couple of quick points.

First, as to the question of whether "look and feel has crashed" or not, I do not think "look and feel" is a very useful term in describing the protection of the external elements, the user interface of a computer software program. It is the kind of term that is so vague that it has the capacity to scare the living daylights out of people, or to confuse them. It can mean anything you want it to mean, or nothing at all, and I do not believe that it is an analytically useful term.

The user interface, on the other hand, whether or not it is the most valuable element of the program, is clearly the program's most communicative and expressive element, from the perspective of the user. It is the only part of the program that

---

<sup>85</sup> 803 F. Supp. 487 (D. Mass. 1992).

speaks to the user and to which the user responds. It is expression, pure and simple. It may or may not be art. It is often useful, but that does not disqualify it from copyright protection because Congress knew that computer programs were useful when it decided to protect them. *CONTU* and the legislative history make that quite clear. But it is speech, it is communication, and to the user it is typically (as long as the program works) the only part of the program the user cares about. The internals only matter if they do not work, or at least not well enough. The user interface is the part of the program the user directly experiences.

If one properly understands and applies *Altai*, the instances in which its efficiency concerns and other external factors will actually dictate (in the merger sense) the design of any meaningful element of the user interface are, I submit, quite rare. You can do almost anything in designing user interface.

In the early days, when I was looking for examples of things that were givens in the design of a spreadsheet, I used to say that obviously any spreadsheet had to have columns and rows, and that you would likely identify one with letters and the other with numbers. That seemed like a safe concession until my friends at Lotus came out with a new product call "Improv," which proved I was wrong. It is a spreadsheet program without geographically defined cells. The cell is no longer the intersection of Row 1 and Column A. It is now "November Sales," and you can move it anywhere on the spreadsheet and it maintains its identity.

So almost nothing you can say about the user interface of a program is a given. Virtually none of it is dictated, in a meaningful way, so the kind of constraints that became so critical in *Altai* are rarely a legitimate concern in an externals case. It does not mean, however, the defendant will not try.

Briefly, with respect to the question of whether copyright is the right legal vehicle, I should first point out that it is not an either/or proposition. As Judge Walker pointed out, there are complementary schemes of protection in place today. On occasions such as this, I like to wear my Spiro Agnew watch. This is not any form of political statement, but simply a reminder of a key case called *In Re Yardley*<sup>86</sup> which established the proposition that a single item can at the same time be protected both by copyright and patent, as this nifty little watch is. Many software programs are too.

So to the extent that people argue that it is either one or the

---

<sup>86</sup> 493 F.2d 1389 (C.C.P.A. 1974).

other, you cannot have both forms of protection, they are simply mistaken. You can have both, provided that each fits appropriately within the statutory framework.

Let's look at the alternatives to copyright.

First, consider trade secret protection. To the extent that what you are trying to protect is a program that is mass marketed, like most application programs, anything the user sees, you are not going to be able to protect as a trade secret. And in the wake of *Sega*, there is a question about how effectively one will even be able to protect the trade secrets contained in what one does not see.

So I don't think that that's an adequate answer.

Neither are patents. Looking at patent protection as the primary way to safeguard software is a lot like chasing flies with a shotgun. Anybody who has dealt with a software patent has seen their potentially fast impact. They have the potential to preclude an entire industry from competing in a given field. Unlike copyright law, which only punishes those who copy, patent law punishes the innocent infringer, along with those who knowingly cross the line. All copyright requires is that you do your own work. It must be original. You can not copy somebody else's work. Under patent law, even if it's your own work, even if you had no idea that the patent was there, even if you built an entire industry around the product during the period when the patent application was secretly pending in the patent office, you still can lose.

So I don't think that that's a satisfactory answer either.

Copyright law seems to work for software. There is some breakage. As Judge Walker pointed out, it is sometimes difficult to apply old doctrine to new technology, but I would suggest to anybody with a historical view of copyright law that this is not the first time this has happened, and it won't be the last.

At the turn of the century there was a similar to-do about player piano rolls. It required Congress and the Supreme Court to go back and forth to decide what the law was going to be. That is just the price of the system.

Copyright law works. This is a vitally important industry. The adequate and effective protection of its intellectual property interests is critical to its survival. I would argue that we not throw out a statutory scheme under which this industry has thrived.

*Professor Rochelle Cooper Dreyfuss*

Those of us who are engaged in teaching law and writing case books tend to approach a new case in two ways. First, we ask whether it's rightly decided and reasoned, but that is only a minor concern. What we really care about is how well a case educates and instructs those in the legal field. *Altai* teaches us extraordinarily well. Not only does *Altai* beautifully lay out the legal issues, but it also does a wonderful job explaining how computers work and demonstrating to students that, it is indeed possible to make highly technical material accessible to non-scientists. Therefore, I would like to thank and congratulate Judge Walker for the *Altai* opinion.

The *Altai* opinion neatly captures the dichotomies that the law is working with in this area; dichotomies that I believe are often insufficiently explored and acknowledged. Teaching *Altai* gives me the opportunity to focus on four such dichotomies: the distinction between patent and copyright; between copying and using; between new and mature industries; and the difference between state and federal protection. These dichotomies exist within the current state of the law, as was accurately demonstrated in the *Altai* decision. I, therefore, thought it would be useful to spend my time speaking about these distinctions.

First, patent versus copyright with regard to computer software. In virtually every discussion of this sort, someone makes the observation that we do not need to worry about copyright law, because patent law provides appropriate protection to functional products, such as computer programs. Why do we hear this said so often? I suspect that it betrays a certain unease with leaving this industry adrift. That is, usually this observation gets made right before or right after the observer explains that copyright does not extend to whatever aspect of programs are under discussion. So why is such an assertion made? Most likely because the speaker knows very well that in this society the principal method for encouraging investment, be it in blackacre, the stock market, or anything else, is the promise that the investor will reap the rewards the investment makes possible. And, as Mr. Clapes told us, for intellectual property such as computer programs, that promise requires that exclusive rights be created by operation of law. After all, there is no physical domination over programs, as there can be physical possession over blackacre. And so, if exclusive rights are needed, they must be created through copyright law, or patent law, or state law.

Will patent protection take up the slack left by denial of copyright protection? I do not think so. If it is true that copyright cannot protect the nonliteral elements of computer programs because they are ideas, then it is difficult for me to believe that patent law will be any more hospitable. It is true that as a statutory matter, patent law does not contain the equivalent of section 102(b) of the Copyright Act in that it does not expressly remove ideas from the ambit of protection. However, as a matter of case law, patent law is no more protective than copyright law. Indeed, it fails to protect ideas for exactly the same reason copyright rejects them. It is just too expensive to give someone with a raw idea the rights over the entire domain of knowledge that the particular idea makes feasible. So, where copyright draws a distinction between ideas and expression, patent draws a line between ideas and application. While the copyright class is off struggling with *Baker v. Selden* and the copyrightability of a book-keeping method, the patent class is grappling with *O'Reily v. Morse* and the question whether Samuel Morse's discovery that the motive power of an electric current may be used to print intelligible characters at any distance is a patentable apparatus (i.e., the telegraph), or an unpatentable principle of nature. The answer, by the way, was an unpatentable principle of nature.

Recently, several pure software patents that are not tied to industrial applications have been issued by the Patent and Trademark Office, some of them covering precisely the sort of material that this conference is about. But it is worth pointing out that so far there have been few challenges to these patents by competitors of the patentee, as opposed to by the commissioner of patents; or in federal courts, as opposed to in the Board of Patent Appeals and Interferences. Thus, we have no real way of knowing how well these patents will hold up, or if they do hold up, how broadly they will be interpreted.

I suspect that if the copyright cases correctly classify nonliteral aspects of programs as ideas, then the patent cases will not reach a very different result. Either, the method for achieving these aspects will be regarded as algorithms that preempt their field—that being patent's way of calling something an idea. Or, the application of the idea will be regarded as very specific to its implementing code, so that the range of equivalents will be very narrow—that being patent's way of saying something has not been infringed. In short, if we persist with the notion that nonliteral elements are ideas, the bottom line for both copyright and patents will be no protection. But, what if we say nonliteral ele-

ments are considered patentable subject matter. It is tempting to assume that patent law would work better than copyright because programs look more like patentable products than copyrightable products, or because people who write computer programs look more like technical nerds than artistes. However, programs are hybrids. The products themselves look like neither patentable subject matter or copyrightable subject matter; the people who write them resemble neither patentees nor copyright holders. It, therefore, becomes tempting to look at patents, rather than copyright, because the term of protection for patents is so much shorter: seventeen years as opposed to the seventy-five that copyright would give to works for hire. But this distinction is equally illusory, for in a fast-paced industry such as computers, seventeen years may as well be seventy-five years. In the year 2010, I doubt people are going to be using programs written in 1993.

In sum, neither the shorter term, nor the look of the product or the creator creates a sufficient argument for patent protection. In fact, in important ways, patent protection is much worse than copyright. To see how, let us move on to the next dichotomy, that between copying and using.

I just finished saying that the perceived difference between patent law and copyright law's protection on protecting ideas is not real. But, there are clear distinctions between these bodies of law, the most being the distinction between copying and using. For instance, once a widget is patented, everyone who uses that widget without permission is an infringer, including any person who independently invents it. That is, someone who creates a widget without seeing the first one or even knowing of its existence is nonetheless barred from using her invention without the authority of the patent holder. In contrast, copyright protects only against copiers. Someone who independently writes identically the same work as a copyrighted work can use his own version free of claims of infringement.

Let us take some time to think about what this means, and what its significance is in social terms. The name of the game here is protecting innovators from being undercut in the marketplace by free riders, without overly raising the price of innovative products or inhibiting downstream users from exploiting new ideas. Which regime does that better, copyright or patent? Both protect against the free rider. If you can not rip-off a protected work, you can not undercut the innovator. But, patent law goes further than copyright. It gives the patentee rights against the entire world. Here is the difference. The absolute protection of



patent law means that so long as there are no close substitutes, the patentee can charge whatever she likes for her product, perhaps more than its cost of development. The copyright holder can not do that. For fact-based works or those works that incorporate standard programming devices, for works whose value lies in efficiency, the copyright holder can not charge whatever she likes. There is a cap on price in copyright because a second user can always do just what the copyright holder did. She can get the facts from the same source, use the same standard devices, work towards the same efficiency objectives, and achieve them in precisely the same way, just so long as there is no copying.

Because the two independent developers now have to compete, the end user will pay something close to the cost of development. The public gets the benefit of competition, something which patent law would never tolerate, and yet the two developers compete on a level playing field because both face the same costs and the same physical constraints, so that there was no free ride. In other words, once the difference between a regime that prohibits using and a regime that prohibits only copying is acknowledged, it is easy to see one of the reasons why copyright is socially preferable to patent law.

To see some other reasons why copyright may be more socially preferable, let us move on to the next distinction: new versus mature industries. Here there are several ramifications. One is that as an industry matures, the demand for legal protection tends to increase. When computer science was new, the thrill of discovery propelled innovation and many advances occurred in academia, which has its own system of reward. But the industry has matured and now advances are mainly achieved through sweat—diligent, time-consuming, money-intensive plugging away. In such an environment, the profit motive becomes more salient and the demand for legal protection against free riders more strident.

The issue then becomes whether the demand should be for copyright or patent protection. I believe that the preference should be for copyright protection because copyright is more suitable to the computer industry's current state of development. At the dawn of this industry, patent protection might have made more sense. Changes happened very fast and the time to recover costs on many introductions was short, so that absolute protection against the entire world is necessary. But, now that the shelf life of introductions is longer, costs can be recovered more slowly and with less social disutility.

Copyright may also be the more available form of protection. Even if it were clear that the nonliteral elements of programs were patentable subject matter, there would still be a question as to whether particular programs merited protection. Whereas copyright protects any original work so long as there is authorship, patents protect only obscure improvements over prior art. When an industry is new, every advance is significantly different from its predecessors and qualifies for protection. But, as an industry matures, there is a leveling off; new introductions are closer to what went before and many will not be different enough to merit patents. Thus, there is an irony here. As the industry matures, it needs more legal protection. The copyright theorists point to patent, but that regime becomes less suitable as the industry matures. There is one more solution that is sometimes suggested here, and that brings me to the final dichotomy: state versus federal protection.

Case law has primarily directed the computer industry to rely on patent law, however, there have been some cases recognizing rights of action under state law. Indeed, the opinion in *Altai* is just such a case. No one is truly comfortable exposing this industry to the blatant free ride. However, one must consider, as a normative matter, whether we are better off substituting for copyright a regime that relies on state law. I would say not, for several reasons.

First, state law is uncertain. A particular fact pattern will give rise to a right of action in some states, but not in others. We wind up with the worst of both worlds: higher prices, limits on access, a chill on downstream use, plus we have no assurance of cost recoupment and, therefore, insufficient motivation to the industry. Indeed, it is worth extending Ms. Braden's lesson. If one goes back and looks at the Federalist papers, one will see that one of the reasons copyright was included in the Constitution was because the Framers thought that a jurisdiction-by-jurisdiction approach simply would not work.

Second, state law is largely common law. No legislative body sits down and thinks about any particular industry to weigh the benefits of protection against costs, to set optimal time limits, or to provide for compulsory licensing when needed. Accordingly, state law is even more poorly tailored to the needs of the computer industry than current copyright law, which at least benefited from material like the CONTU report. For instance, we talked about compatibility and standardization in user interfaces. One could imagine that different positions are taken in the law

on each of these issues. We might want freer public access to achieve compatibility, but greater protection for creators of user interfaces. Congress could hold hearings, listen to the views of experts, review data from empirical studies, and then come up with whatever fine distinctions were necessary. Would states do as well? Some states might, but certainly not all.

Finally, much of state law may not survive federal preemption. A number of years ago the Supreme Court decided a case called *Bonito Boats v. Thundercraft Boats*, which preempted a Florida law prohibiting a particular method for copying boat hulls, not under copyright law, but rather by applying patent law. Essentially, the Court took the position that if an invention does not merit patent protection, then the Constitution requires that the invention be in the public domain; states simply have no role in protecting it. Now, I have to be quick to add here that the Supreme Court is very enthusiastic these days about contractual rights, so a case like *Altai*, which involved the confidential duties of an employee, is surely good law. However, not all computer cases involve contractual or quasi-contractual rights. Therefore, I do not believe the computer industry should look for enduring safeguards in state law anymore than it should look for them in patent law. And I mean "should" both as a descriptive and a normative matter. But that is not to say that copyright law has nothing to learn from patent law, or even from state law. Both patent law and state law (patent in particular), have had more experience with functional products than copyright has had. Thus, courts could usefully adapt things like patent's approach to remedies and defenses in deciding copyright cases. But, absent enactment of a *sui generis* regime, the federal anticopying rules of copyright are surely better than the alternatives.

And so in a sense, I leave you where you started; struggling to understand copyright in a way that protects the significant investment that this industry makes in the nonliteral elements of programs, without destroying the trajectory of development that functional products traditionally tend to follow. For my part, I am very much in favor of holding virtually all aspects of original programs copyrightable. I would rely on independent invention, on copyright's fair use, on patent's misuse defense, and perhaps new compulsory licensing provisions to keep prices reasonable and to insure adequate levels of innovation.

