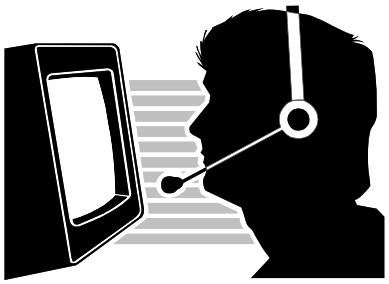
A Comprehensive Current Analysis of Software "Look and Feel" Protection



David L. Hayes, Esq.*
FENWICK & WEST LLP

I. INTRODUCTION

The United States software industry has experienced phenomenal growth since its inception, and remains one of the few technical industries in which the United States demonstrably leads other countries. To maintain such growth and to protect market position, software companies continue to place strong emphasis on intellectual property rights. Many cases decided soon after passage by Congress in 1980 of an amendment to the copyright law explicitly bringing computer programs within the ambit of copyright protection established that copyright extends to the "literal" elements – the source code and object code – of computer programs of all types.[1]

In recent years, most of the copyright infringement cases relating to software have sought to extend copyright protection to "nonliteral" elements of computer programs. Early on, the phrase "look and feel" – though not a copyright term of art – was coined as a label for the various nonliteral elements of a computer program.[2] The "look" of a program includes its demonstrative audiovisual elements – its screen displays, visible portions of the user interface and other visual and aural elements of output produced by the program. The "feel" of a program includes the dynamic, operational flow of the program, its keystrokes and other means for invoking functions, and the general recognizable "style" of operation the program presents to the user. In many instances, the "look" and "feel" categories overlap. "Feel" has also been used to

---

* Chairman of Intellectual Property Practice Group, Fenwick & West LLP, Palo Alto, California.

1 See, e.g., CMS Software Design Sys., Inc. v. Information Designs, Inc., 785 F.2d 1246 (5th Cir. 1986) (source code); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), cert. dismissed, 464 U.S. 1033 (1984) (source and object code of an operating system); Williams Electronics, Inc. v. Artic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982) (object code of a video game program).

2 See Russo & Derwin, "Copyright in the 'Look and Feel' of Computer Software," 2 The Computer Lawyer 1 (February 1985).

refer more generally to the nonliteral "structure, sequence and organization"[3] of a program – both its static modular structure and organization, and its dynamic operational sequence of program control and data flow. In addition, the "feel" of a program has sometimes been used to encompass other nonvisible elements such as file formats, data structures, commands, and system calls.

The "look and feel" cases decided during the last five years can be collectively characterized – in the recent words of the Fifth Circuit – as "in a state of creative ferment."[4] On the "look" side, the courts have sought to define what elements of user interfaces are so commonplace or standard in the industry, or so merge with their inherent functional role in the interface, as to be unprotectable. On the "feel" side, the courts have struggled to draw the line between where unprotectable function[5] per se ends, and nonliteral, "structural" expression begins, particularly with respect to those elements of a computer program – such as modular structure, dynamic behavior, logic and data flow, keystrokes, menu command structure, system calls, file formats, data structures, and functional features – that are not entirely visible and concrete, either in the program's code or its screen displays.

Out of this struggle by the courts has come a series of complicated and sometimes confusing decisions. In the last few years, a number of new tests have emerged for judging what constitutes copyrightable "expression" and how one judges whether an allegedly infringing work is "substantially similar" to a copyrighted work.[6] These tests have been motivated by the difficulty of applying well-known copyright doctrines, originally developed for traditional literary and artistic works such as plays, novels, art and musical works, to computer programs, which are inherently functional and technical.[7]

---

3 Whelan Assocs. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1224 n.1 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

4 Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1341 (5th Cir. 1994), supplemental opinion & reh'g en banc denied, 46 F.3d 408 (5th Cir. 1995).

5 Section 102(b) of the copyright statute provides: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b). This provision codifies what has generally been referred to as the "idea/expression" dichotomy. See, e.g., Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 703 (2d Cir. 1992).

6 To establish infringement, a copyright owner must prove ownership of a valid copyright and copying. Copying may be proved either by direct evidence or, as in most cases, by indirect evidence showing that the defendant had access to the plaintiff's copyrighted work and that the defendant's work is substantially similar to the plaintiff's copyrighted work. See, e.g., Computer Associates, 982 F.2d at 701.

7 "The essentially utilitarian nature of a computer program further complicates the task of distilling its idea from its expression. In order to describe both computational processes and abstract ideas, its content 'combines creative and technical expression.' ... Thus, compared to aesthetic works, computer programs hover even more closely to the elusive boundary line described in § 102(b)." Id. at 704 (citations omitted).

Although the tests are articulated in different ways, the goal of each of them is generally to identify uncopyrightable elements that must be removed from the allegedly copied nonliteral "expression" before infringement is adjudicated by application of a similarity test. The clear trend of the decisions is to label more and more elements as unprotectable based on an increasing number of grounds, thus often tending to narrow the scope of protection afforded by the copyrights at issue in the cases.

To date, however, the decisions have not clearly defined the most critical aspect of judging whether two programs are substantially similar: if some of the similar elements of a program are copyrightable and other similar elements are not, how are the copyrighted work and the allegedly infringing work to be compared to adjudicate substantial similarity? A number of possibilities exist. Having identified and removed the uncopyrightable elements from the plaintiff's work, should one then compare the similar elements of the two programs as a collection of distinct elements without regard to the possible relationships among them – such as arrangement, layout, and dynamic sequence? Or, because the relevant totality of protectable elements may be greater than the sum of its parts, should one compare the similar elements in the defendant's work[8] to some larger totality in the plaintiff's work? That totality could be comprised of (1) all of the similar copyrightable elements, (2) all of the potentially copyrightable elements (i.e., both similar and dissimilar copyrightable elements), or (3) the entire work.[9]

How one treats individual elements that may be "unprotectable" standing alone when judging overall similarity between a plaintiff's and defendant's *works* can greatly affect the outcome of the case. The plaintiff may, for example, have put together a large collection of individually unprotectable elements into an arrangement, sequence or collection that, taken as a whole, is original and potentially protectable. Throwing out all elements from the work that may be individually unprotectable to adjudicate similarity risks eliminating such arrangement, sequence or collection from the scope of copyright, a result which is inconsistent with a long line of cases protecting compilations – works which are often made up of individually unprotectable elements.[10]

The look and feel cases often exhibit a difficult tension in the policy underlying the copyright law as it applies to computer programs, which are technological works that are designed to serve a functional, as opposed to purely aesthetic, purpose. In judging the scope of

---

[8] It is an established rule that the defendant may not avoid a claim of infringement by showing how much protectable expression and/or other elements the defendant's work contains that are not similar to the plaintiff's work. Rather, the infringement focus must be on the extent to which the defendant copied protectable elements found in the plaintiff's work, and whether the copied protectable elements are a qualitatively substantial part of the plaintiff's work. See, e.g., id. at 710, 714.

[9] See Zimmerman, "Substantial Similarity of Computer Programs After Brown Bag," 9 The Computer Lawyer 6, 14 (July 1992).

[10] See., e.g., Feist Publications, Inc. v. Rural Telephone Serv. Co., 499 U.S. 340 (1991); Harper House, Inc. v. Thomas Nelson, Inc., 889 F.2d 197 (9th Cir. 1989).

copyright, one must exercise care not to lose sight of the forest for the trees. On the other hand, there is a natural tendency to resist extending protection to elements of an interface or other portion of a computer program that may not be original to the plaintiff, may have become standard in the industry, or may be necessary or desirable to maintain consistency or compatibility across computer programs from various vendors.

In sum, the courts have not clearly answered the question of once having identified and "filtered" the uncopyrightable elements of the "look and feel" of the plaintiff's program, what does one do with what's left? Curiously, the decisions to date have exhibited different trends in results reached on the point: while the "look" cases have tended to narrow what visible aspects of a user interface are protectable, a number of the "feel" cases, at least at the district court level, have tended to expand what is protectable in the non-visual, nonliteral aspects of a program (such as keystrokes and embodied techniques and features), although in the last few years several decisions have rejected claims for protection of various functional aspects of a program such as commands.

In addition, a number of other trends are evident in both the "look" and the "feel" cases:

- There is a greater tendency to recognize certain "externalities" – such as functionality, compatibility, industry standardization, and commonplace features – as limiting the scope of what elements of a program are protectable.

- A majority of courts now reject the early approach adopted in the Whelan decision[11] for determining what is protectable in a computer program, insofar as that case defined but a single "idea" embodied in a computer program for applying the idea/expression distinction: "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."[12] Most courts have now explicitly recognized that a computer program may contain many "ideas," and those ideas must be defined at various levels of abstraction in order to sift out protectable from unprotectable elements of a program at those levels of abstraction. This broader approach to defining "ideas" narrows what the single-idea approach of the Whelan case would relegate to the realm of "expression."

- The courts are abandoning sole use of the "lay observer" test to judge substantial similarity, and are increasingly relying on expert testimony to aid both in the dissection of a work into protectable and unprotectable elements, and in substantial similarity.

- The opinions exhibit a heightened sensitivity to the inherent balance of the copyright law between rewarding authors as a motivation to create new copyrighted works and

---

[11] Whelan Assocs. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

[12] Id. at 1236 (emphasis omitted).

the free flow of information and use of ideas. Many opinions explicitly recognize the potential effect on competition that broad copyright protection for computer programs can have. Several recent decisions have applied the fair use doctrine more broadly to claims of copying of functional elements of a program for compatibility reasons.

- Where many or most of the alleged similarities in a computer program are either not protectable or licensed, many courts are tending to require a very high degree of similarity – expressed by the Ninth Circuit in the Apple v. Microsoft case as "virtual identity"[13] – to find infringement.

This paper analyzes the current state of "look and feel" protection for computer programs. Beginning with the early business software cases from the mid-1980s, it gives a comprehensive exegesis of the various tests that the courts have articulated for judging what is protectable and whether there is infringement, catalogs what specific elements of user interfaces and program "behavior" have been held to be protectable and unprotectable, and discusses the trends of legal analysis evidenced in these cases. For convenience of organization, the "look" cases are analyzed separately from the "feel" cases, but the cases are compared and contrasted in an attempt to synthesize some general principles from a very complex body of case law that sometimes seems contradictory on various points.

II. ANALYSIS OF THE "LOOK" CASES

A. EARLY BUSINESS SOFTWARE CASES

The earliest cases to protect screen displays produced by computer programs were the video games cases brought in the late 1970s and early 1980s, which treated the fanciful screen displays of the games produced by the computer program as "audiovisual works"[14] that constituted a copyrightable work separate from the "literary work" comprising the computer

---

[13] "When the range of protectable and unauthorized expression is narrow, the appropriate standard for illicit copying is virtual identity." Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1439 (9th Cir. 1994), cert. denied, 115 S. Ct. 1176 (1995). This standard of similarity is akin to that imposed by the Ninth Circuit for compilations: "[C]opyright infringement of compilations consisting largely of uncopyrightable elements should not be found in the absence of 'bodily appropriation of expression.'" Harper House, Inc. v. Thomas Nelson, Inc., 889 F.2d 197, 205 (9th Cir. 1989).

[14] "Audiovisual works" are defined by the copyright statute as "works that consist of a series of related images which are intrinsically intended to be shown by the use of machines or devices such as projectors, viewers, or electronic equipment, together with accompanying sounds, if any, regardless of the nature of the material objects, such as films or tapes, in which such works are embodied." 17 U.S.C. § 101.

code itself.[15] Buoyed by these cases, owners of business software began in the mid-1980s to attempt to protect the look and feel of the user interfaces and other aspects of the screen displays produced by such business software through copyright.

1. The Softklone Case

One of the earliest successful cases was Digital Communications Assocs. v. Softklone Distributing Corp.,[16] which held that the main "status screen" of the Crosstalk XVI communications program was protectable as a compilation of command terms. The status screen contained a number of parameter/command terms grouped under various descriptive headings. Each command term reflected the setting of a parameter affecting the transmission of data through a communications channel. Two letters of each parameter/command term were capitalized and highlighted on the screen. By typing those two letters, the user could invoke the corresponding command to change a parameter. The defendant produced a competing communications program that had a status screen virtually identical to the plaintiff's status screen.

The court held that the status screen was copyrightable and that the defendant had infringed. The court found that the particular arrangement and grouping of the parameter/command terms on the screen, and the highlighting and capitalizing of two specific letters of each term, constituted copyrightable expression which the court found to evidence "considerable stylistic creativity and authorship above and beyond the ideas embodied in the status screen."[17] The court explicitly rejected defense counsel's argument that the need for standardization in the computer industry should allow the defendant more latitude in using the same arrangement of status screen commands as the Crosstalk program, which had become a standard.

2. The CAMS Case

A second case decided two years later, Manufacturers Technologies, Inc. v. CAMS, Inc.,[18] also involved screen displays consisting entirely of textual material, but went one step beyond the Softklone case, in that the defendant had not copied the plaintiff's screens virtually verbatim. The plaintiff was the owner of a computer program designed to enable the user to estimate the cost of machining a manufactured part. The program led the user through the estimation via a sequence of screens asking for various information in a predefined, logical order. The defendant marketed a competing cost estimation program.

---

[15] See, e.g., Stern Electronics, Inc. v. Kaufman, 669 F.2d 852 (2d Cir. 1982); Williams Electronics, Inc. v. Artic Int'l, 685 F.2d 870 (3d Cir. 1982); Midway Mfg. Co. v. Strohon, 564 F. Supp. 741 (N.D. Ill. 1983).

[16] 659 F. Supp. 449 (N.D. Ga. 1987).

[17] Id. at 460.

[18] 706 F. Supp. 984 (D. Conn. 1989).

The court concluded that both the "external" and certain "internal" aspects of the plaintiff's screen displays were copyrightable and had been copied by the defendant. The "external" aspects were the sequence of screen displays that led the user's thought processes through a number of manufacturing and engineering decisions, which the court found to communicate to the user the plaintiff's view of how a cost estimate should be created. The court found that the external flow and sequencing of the screens was not dictated solely by functional considerations and therefore constituted expression.[19]

The court also found that the following various "internal" aspects of individual screen displays were copyrightable and had been copied by the defendant in whole or in substantial part:

- The selection and placement of certain identifying information related to the estimate directly under the underscored title of the program (the operation or department at the top left of the screen, the tool number selected at the top right of the screen, and the name of the selected tool in the top center of the screen).

- The layout and choice of nine job identification parameters on an identification screen, several of which were redundant.[20]

The court, however, found the following aspects of the plaintiff's user interface not to be copyrightable – many of which related to the "feel" or other non-visual aspects of the user interface:

- The use of certain keystrokes for navigation among items in menus (space bar to move the cursor down a list, backspace key to move up a list, return key to activate a selected function, and use of number selection to change or edit an entry). The court found that the hardware on which the program was designed to run constrained the type of key assignments that could be used for navigation. This is one of the first decisions to explicitly recognize hardware constraints as limiting the scope of nonliteral elements subject to copyright.

- The use of menus to navigate from screen to screen and among various components of the program.

- Display of information in a two-column format and with upper and lower case letters.

- The choice and display of certain specific items of data relating to specific tooling operations, because those items were determined by the nature of the machining industry itself. Thus, if the industry to which a program relates imposes certain

---

[19] Id. at 994; see also Broderbund Software Inc. v. Unison World, Inc., 648 F. Supp. 1127 (N.D. Cal. 1986) (sequence of screens for creating banners, cards and signs constituted copyrightable expression).

[20] CAMS, 706 F. Supp. at 994-95.

> "externalities" that dictate certain forms of expression in the program, those forms are not copyrightable.[21]

As has become increasingly true in the cases over time, the court relied heavily on expert testimony to understand the similarities between the plaintiff's and defendant's programs and whether such similarities were driven by functional concerns or industry externalities.

B. THE DATA EAST v. EPYX CASE

A seminal case, Data East USA, Inc. v. Epyx, Inc.,[22] was decided by the Ninth Circuit in 1988. Although a video game case, this case laid the groundwork for the analytic dissection of user interfaces of business software to remove uncopyrightable elements that the Ninth Circuit would later establish in the Brown Bag case,[23] discussed below. The plaintiff in Data East was the owner of the copyright in a video game called "Karate Champ." The defendant distributed its own karate game called "World Karate Championship," which the plaintiff alleged was infringing. The district court agreed, citing numerous similarities between the two games.

The Ninth Circuit reversed, finding that all the similarities cited by the district court related to unprotectable elements of the plaintiff's karate game. "To determine whether similarities result from unprotectible expression, analytic dissection of similarities may be performed. If this demonstrates that all similarities in expression arise from use of common ideas, then no substantial similarity can be found."[24] The court concluded that all similarities cited by the district court[25] "necessarily follow from the idea of a martial arts karate combat game, or are inseparable from, indispensable to, or even standard treatment of the idea of the karate sport. As such, they are not protectible."[26]

The court found that the only protectable expression in the plaintiff's game related to the scoreboard and the background scenes, and these elements were in fact dissimilar in the

---

[21] Id. at 995-96; accord Plains Cotton Cooperative v. Goodpasture Computer Serv., 807 F.2d 1256 (5th Cir.), cert. denied, 484 U.S. 821 (1987) (similarities in the plaintiff's and defendant's order-management computer programs were dictated by the externalities of the cotton market).

[22] 862 F.2d 204 (9th Cir. 1988).

[23] Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir.), cert. denied, 113 S. Ct. 198 (1992).

[24] 862 F.2d at 208.

[25] The district court found that each game had fourteen moves, a two-player option, a one-player option, forward and backward somersault moves and about-face moves, a squatting reverse punch in which the heel is not on the ground, an upper-lunge punch, a back-foot sweep, a jumping sidekick, a low kick, a walk-backwards position, changing background scenes, 30-second countdown rounds, and a single referee. Id. at 209.

[26] Id.

defendant's game. Because there were no similarities of protectable expression, the district court's grant of an injunction on behalf of the plaintiff was reversed.[27]

The Ninth Circuit had no occasion in Data East to opine on how one judges substantial similarity when there are similarities in both protectable and unprotectable elements. The case did establish, however, that when analytic dissection establishes that all similarities relate to unprotected elements, summary judgment for the defendant is appropriate.

C. THE PAPERBACK CASE

In 1990, Judge Robert Keeton of the District of Massachusetts decided the first of two very important copyright infringement cases brought by Lotus Development Corporation on the copyrights in Lotus' spreadsheet program, 1-2-3. The first of these cases, Lotus Development Corp. v. Paperback Software Int'l,[28] was largely a "look" case brought by Lotus against Paperback Software, the maker of a "clone" of 1-2-3.

The user interface of the plaintiff's spreadsheet product consisted in part of a highlighted screen display resembling an "L" rotated ninety degrees clockwise with letters across the top to designate columns, and numbers down the left side to designate rows, the intersection of which formed a grid of "cells" into which numbers or formulas could be entered to accomplish spreadsheet calculations. The 1-2-3 user interface also had a "two-line moving-cursor menu" at the top of the screen, which presented the user with a menu of command terms. The menu was called up to the screen by pressing the slash ("/") key. The top line of the two-line menu contained a series of command terms.[29] Each command term could be selected by pressing the left or right cursor keys until that command term was highlighted, then pressing enter, or by pressing the first letter of the command term on the keyboard.

The second line of the menu displayed a "long prompt," which in many cases consisted of a textual description of the currently highlighted command term, and in other cases provided a list of the menu command subchoices that would become available if the highlighted command was chosen.

The 1-2-3 product also contained a "macro" facility, enabling the user to store a sequence of command terms as a "macroinstruction" (or simply "macro") and then, with one command stroke that invokes the macro, to cause the computer to execute the stored sequence of commands. The commands of a macro could be menu choices, keyboard commands (such as function keys or cursor keys), or certain special macro commands. Because macros could contain menu choices, the exact hierarchy of the menu system constituted "a fundamental part of the functionality of the macros."[30]

---

[27] Id. at 209-10.

[28] 740 F. Supp. 37 (D. Mass. 1990).

[29] For example, the top line of the first, or main, menu in 1-2-3 read: "Worksheet Range Copy Move File Graph Data Quit". Id. at 64.

[30] Id. at 65.

The user interface of the defendant's "clone" product was virtually identical to that of 1-2-3, except that the two-line moving-cursor menus were at the bottom of the screen, the defendant's product had numbers associated with each command term that could be used as an alternative means to invoke the command, and certain new commands not contained in 1-2-3 had been added to the menu hierarchy at various points. The defendant argued, however, that the user interface of 1-2-3 was not copyrightable because it or various elements of it were a "useful article,"[31] a blank form, and an unprotectable computer language, and for various other reasons.[32]

The court rejected all of these arguments, and concluded that the user interface of 1-2-3, taken as a whole, was copyrightable. The court acknowledged that utilitarian aspects of a work per se are not copyrightable, and that even "the expression of the idea is not copyrightable if the expression does no more than embody elements of the idea that are functional in the utilitarian sense."[33] However, if "the expression of an idea has elements that go beyond all functional elements of the idea itself, and beyond the obvious, and if there are numerous other ways of expressing the non-copyrightable idea, then those elements of expression, if original and substantial, are copyrightable."[34]

The court formulated the following three-step test for determining what elements of a work constitute copyrightable expression that must be compared under the substantial similarity test to adjudicate infringement:

> FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation – some conception or definition of the "idea" – for the purpose of distinguishing between the idea and its expression.[35]

---

[31] The copyright statute defines a "useful article" as one "having an intrinsic utilitarian function," and provides that copyright protects such articles only to the extent that they contain elements "that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article." 17 U.S.C. § 101.

[32] 740 F. Supp. at 54-58, 71-73.

[33] Id. at 58.

[34] Id. at 59.

[35] Id. at 60 (emphasis in original). This first step of the test invokes the familiar "abstractions test," under which a work is conceptualized at a number of levels of abstraction, from the most general to the most concrete. It was first formulated by Judge Learned Hand in Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), cert. denied, 282 U.S. 902 (1931): "Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are

> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of that idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of that idea.
>
> THIRD, having identified elements of expression not essential to every expression of the idea, the decisionmaker must focus on whether those elements are a substantial part of the allegedly copyrightable "work."[36]

Applying the first step of the test to Lotus 1-2-3, the court defined the idea underlying the program at the highest level of abstraction as "the electronic spreadsheet," and held that such an idea is not copyrightable.[37] The court concluded, however, that expression could exist at a lower level of abstraction – in particular, at the level of the "user interface," which Lotus defined to include "the menus (and their structure and organization), the long prompts, the screens on which they appear, the function key assignments, [and] the macro commands and language."[38] The court concluded that, because there are many different user interfaces that could implement an electronic spreadsheet, the second step of the test was satisfied. Finally, the court found the third step of the test satisfied because it concluded that the "user interface of 1-2-3 is its most unique element, and is the aspect that has made 1-2-3 so popular."[39]

Accordingly, the court concluded that the 1-2-3 interface, and in particular its "menu structure, taken as a whole – including the choice of command terms, the structure and order of those terms, their presentation on the screen, and the long prompts –" was copyrightable.[40] Although the court held that the 1-2-3 interface, at least when taken as a whole, was copyrightable, the court also ruled that the following specific elements within that interface were not copyrightable:

- The two-line moving cursor menu format itself.
- The "rotated L" screen display that formed the basic spreadsheet cell grid.
- The use of the slash key ("/") to invoke the menu command system, because of the limited number of keys that are logically available to serve this function.

---

no longer protected, since otherwise the playwright could prevent the use of his 'ideas,' to which, apart from their expression, his property is never extended. Nobody has ever been able to fix that boundary, and nobody ever can."

[36] Id. at 61.

[37] Id. at 65.

[38] Id. at 63.

[39] Id. at 68.

[40] Id. (emphasis added).

- The use of the "+" key to indicate addition, the "-" key to indicate subtraction, the "*" key to indicate multiplication, and the "/" key to indicate division within formulas stored in a cell, and the use of the "enter" key to store keystroke entries into the cells. The court found the use of such keys to be either "essential to every expression of an electronic spreadsheet" or at least "obvious."[41]

Having established the copyrightability of the 1-2-3 user interface, the court then turned to the question of copying and substantial similarity. The court ruled that no issue remained for a jury on this point, because the defendant had admitted copying the 1-2-3 user interface virtually verbatim, and in any event the similarities between the interfaces of the plaintiff's and the defendant's products were "overwhelming and pervasive."[42] The two products had the same menu structure, and the user interfaces functioned the same, "keystroke for keystroke."[43]

The court did note some differences in other elements of the interface, relating to the appearance of the start-up screens, the placement on the screen of the menu lines, the exact wording of the long prompts, the organization of the help screens, the increased width of the defendant's screens, and the ability of the defendant's product to hide certain columns. The court concluded, however, that these differences were not sufficient to keep the two interfaces from looking "substantially, indeed, strikingly, similar,"[44] and that in any event one should not focus on differences between the allegedly infringing and the allegedly infringed work when a qualitatively substantial part of the allegedly infringed work has been copied.

The court noted that these similarities were overwhelming from "the perspective of both an expert and an ordinary observer,"[45] suggesting (although not specifically holding) that expert testimony is relevant to the issue of substantial similarity. The court also rejected the defendant's argument that it had to copy the menu structure of 1-2-3 in order to create a program that would be completely compatible with 1-2-3, particularly in its ability to read data files created using 1-2-3. The court found that, as a factual matter, the defendant could have achieved such compatibility using a macro conversion facility rather than copying, and that in any event "the desire to achieve 'compatibility' or 'standardization' cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual 'work.'"[46]

Although Judge Keeton's opinion contains a very long, if somewhat prolix, exegesis concerning the application of copyright principles to computer user interfaces, the usefulness of the opinion is limited by the following:

---

41 Id. at 66.

42 Id. at 68.

43 Id. at 69.

44 Id. at 70.

45 Id. (emphasis added).

46 Id. at 69.

- Judge Keeton's three-step test is only a test for what constitutes copyrightable elements in a user interface. The test says nothing about what one does with the elements that have been determined to be uncopyrightable in judging substantial similarity, or indeed really anything else about how one is to compare the allegedly infringing work to the allegedly infringed work. Because of the unique posture of the case, in which the defendant admitted copying and the user interfaces between the two products at issue were in many ways virtually verbatim in both "look" and "feel," Judge Keeton was not required to opine on the more difficult issue of how the substantial similarity comparison is to be carried out. Because the user interface of the defendant's product was strikingly similar to that of the plaintiff's product – both element-by-element and overall – the court was not required to specify in detail how two computer user interfaces are to be compared for substantial similarity.

- Even with respect to issues of copyrightability, Judge Keeton's test is indeterminate. The test specifies that the court must define the various levels of abstraction one can conceive for defining the "idea" underlying the copyrighted work and choose one level for drawing the legal line between idea and expression, but gives no rule and little guidance as to how that line is to be chosen. Moreover, the test does not articulate criteria for determining whether any particular element of a user interface is "essential" to the expression of the idea chosen.

D. THE APPLE v. MICROSOFT CASE

One of the most closely watched look and feel cases was the seminal case of Apple Computer, Inc. v. Microsoft Corp. Apple first brought its complaint in 1988, alleging that the user interface of Version 2.03 (and later Version 3.0) of Microsoft's "Windows" system software product, and the user interface of Hewlett Packard's "New Wave" product (which runs under Windows and enhances both the visual and functional elements of the Windows interface), infringe Apple's copyrights in the user interface of its "Lisa" and "Macintosh" computer operating systems.

Neither the "Windows" product nor the "New Wave" product was a "clone" of the Macintosh user interface. Nevertheless, the interfaces of the various products shared many elements in common – overlapping windows, scroll bars, pull-down menus, manipulable and movable icons, and many other graphical user interface (GUI) elements – giving the interfaces certain definite and clearly articulable similarities when viewed as a whole. Yet many of these similarities resulted from the presence of individual elements in the interfaces that the court ultimately found not to be original to Apple or to be "unprotectable" standing alone because subject to various "limiting doctrines" of copyright law such as merger, scenes a faire, and functionality, or because they had been licensed to Microsoft by Apple.

Both the district court and the Ninth Circuit ultimately resolved the tension between the risk of losing the forest for the trees and the risk of overprotection by adopting a test that makes the standard of infringement for the works as a whole turn on the protectability of the individual elements comprising the whole. In particular, both courts in effect held that where a work is entitled to only limited copyright protection – either because it is consists largely of elements that are either licensed or unprotectable, or because the alleged similarities are capable of only a

narrow range of expression – the standard of infringement to be applied is a "virtual identity" standard, rather than the more expansive, traditional "substantial similarity" standard.

Remarkably, however, despite a total of six years of litigation including the appeal, neither the district court nor the Ninth Circuit ever resolved – because of a stipulation between Apple and the defendants that the works in suit were not virtually identical – the ultimate question of real interest in this (and all) look and feel cases: in applying the standard of infringement to the works as a whole, does one remove or include the elements that have been found to be "unprotectable" standing alone?

Despite the case's failure to resolve how specifically to apply the standard of infringement to the works as a whole, the district court's various decisions in the case and the Ninth Circuit's opinion on appeal ultimately spell out comprehensive, and complex, tests for categorizing the allegedly similar elements of a graphical user interface, filtering out "unprotectable" elements, and deciding which standard of infringement to apply – substantial similarity or virtual identity – to the works as a whole. Although these cases have been widely cited for various points, the specific detailed tests that they developed have not been widely relied on in other look and feel cases, perhaps because of their complexity and because of the unique factual circumstances of the case, as detailed below.

To fully understand the Ninth Circuit's opinion affirming the district court, it is necessary to first give a detailed analysis of the case and the district court's many opinions. Accordingly, the next section analyzes the history of the case in some detail, and then depicts the test ultimately adopted by the district court in flow chart form. The following section then analyzes in detail the Ninth Circuit's opinion on appeal, and also depicts the Ninth Circuit's test in flow chart form. The similarities and differences between the district court's test and the Ninth Circuit's slight reformulation of the test are discussed.

ANALYSIS OF THE DISTRICT COURT'S DECISIONS

1. The License Between Apple and Microsoft

The course of the legal analysis in the Apple case was charted from the beginning in large part by a license agreement between Apple and Microsoft. Early rulings in the case interpreting the meaning of that license forced the district court to focus on individual display elements comprising the works at issue, thereby significantly narrowing the scope of protection Apple was able to assert against the defendants.

In 1985, Apple executed an agreement with Microsoft granting to Microsoft a license to use "certain visual displays generated by ... 'Microsoft Windows Version 1.0'" and five named applications programs "in current and future software products."[47] Version 1.0 of the Windows product displayed multiple windows open on the screen in a "tiled" fashion (connected together in a fixed pattern of adjacent tiles such that all open windows were visible simultaneously) and confined the display of icons to a defined area at the bottom of the screen. Version 2.03 of

[47] Apple Computer, Inc. v. Microsoft Corp., 717 F. Supp. 1428, 1430 (N.D. Cal. 1989).

Windows contained "overlapping" windows, which allowed windows to be stacked on top of one another and moved around individually on the screen at will. In addition, Version 2.03 allowed icons to be moved and positioned freely around the screen. Because both of these features of Version 2.03 are present in the Macintosh user interface, the effect of these changes was to cause Version 2.03 (and later Version 3.0) to have an overall look and feel much more similar to that of the Macintosh user interface than did Version 1.0 of Windows.

Microsoft argued that the license agreement granted to Microsoft a right to use individual screen display elements appearing in Windows Version 1.0 in any subsequent product, and in any combination or arrangement. Based on the license, Microsoft argued that, in making any comparison of the two works, the court must first excise all licensed displays and compare only those remaining displays, to the extent copyrightable, not found in Windows 1.0 (and therefore not covered by the license). Microsoft further argued that all displays in Windows 2.03 not found in Version 1.0 were uncopyrightable, and there was therefore no infringement.[48]

Apple contended that the 1985 agreement was only "a license of the interface of Windows Version 1.0 as a whole, not a license of broken out 'elements' which Microsoft could use to create a different interface more similar to that of the Macintosh."[49] Apple argued that, because Microsoft had created in Version 2.03 a product that was more similar overall to the Macintosh user interface than was Version 1.0, Microsoft had taken itself outside the scope of the license agreement, and Version 2.03 was therefore unlicensed and infringing of Apple's copyrights. Apple argued that the court should compare the entire user interface of Windows 2.03 against the entire Macintosh user interface for substantial similarity.

In an early decision in the case, the court (Judge Schwarzer) rejected Apple's arguments and accepted Microsoft's arguments as to the interpretation of the 1985 license. The court found that the agreement permitted Microsoft to use individual visual displays found in Version 1.0 in any way it chose, and that the license was not intended "as a product license restricting Microsoft and its licensees to the use of the Windows 1.0 interface as a whole."[50] This ruling concerning the scope of Microsoft's rights under the license agreement seems to have influenced the court in all subsequent decisions to focus its analysis largely on individual allegedly similar screen elements on an element-by-element basis.

After a review of similarities between the Windows and Macintosh user interfaces submitted by Apple, Judge Schwarzer concluded that all displays giving rise to such similarities were licensed under the 1985 Agreement, except certain displays relating to the storage of icons anywhere on the screen rather than just at the bottom of the screen, the display of the icon's name below the icon, and changes in visual displays necessary to implement the overlapping window system used in Windows Version 2.03.[51]

---

[48] Id. at 1429-30.

[49] Id.

[50] Id. at 1430-31.

[51] Id. at 1433-35.

Having issued its ruling on the meaning of the license agreement, however, the court issued confusing signals as to how the remaining unlicensed displays in Version 2.03 were to be treated in judging infringement. At one point in its opinion, the court noted:

> It is, of course, true, as Apple argues, that in determining whether an audiovisual work infringes, the work must be viewed as a whole. But where a work includes licensed features as well as unlicensed features, infringement depends on whether the unlicensed features are entitled to protection; licensed features are treated as being in the public domain.[52]

Although such language might suggest that the licensed displays would form no part of the comparison for substantial similarity, the court stated as follows in its conclusion to the opinion: "The Court will therefore now proceed to determine whether the use of those unlicensed visual displays in combination with licensed visual displays infringes Apple's audiovisual copyrights."[53] Thus, at this juncture in the litigation, Judge Schwarzer left open the possibility that even the licensed displays might form part of the comparison for substantial similarity, if the combination of those displays with other, unlicensed (and, presumably, protectable) displays was substantially similar to the combination, or totality, of the displays in the Macintosh user interface. As discussed below, this difficult issue of the treatment of licensed displays continued to vex the court throughout the remainder of the litigation.

2. The Evolving Test for Copyrightability and Substantial Similarity

Not long after the opinion concerning the scope of the 1985 license was issued, the Apple case was reassigned from Judge Schwarzer to Judge Walker. Judge Walker then issued a series of decisions, in response to motions for summary judgment brought by all parties in the case, grappling with the issues of which unlicensed visual displays in Windows were copyrightable and which were not, and how a work containing both licensed and unlicensed displays – as well as copyrightable and uncopyrightable displays – was to be compared to the plaintiff's copyrighted work in adjudicating infringement. Judge Walker's struggle with these issues is a paradigmatic example of the extreme difficulty courts have faced in deciding how one should properly treat elements in an interface which may be unprotectable standing alone, but may nevertheless form a part of a larger arrangement, selection, sequence, or layout that may constitute expression.

In his earliest opinion in the case, issued in March of 1991, Judge Walker stated that he felt bound by the approach adopted by Judge Schwarzer of looking at discrete visual displays:

> Implicit in Judge Schwarzer's approach to the case is a rejection of Apple's fundamental contention that the "total concept and feel" of the Macintosh graphic user interface is protectible expression. Rather, Judge Schwarzer's approach appears to have been to exclude licensed visual displays prior to applying the substantial similarity of idea and expression tests. The undersigned has

---

[52] Id. at 1432.

[53] Id. at 1435 (emphasis added).

> considered a different approach to the litigation from that adopted by Judge Schwarzer, one that would not begin by an attempt to parse the visual displays of the Macintosh system software. However appealing such an approach might seem in the abstract, the 1985 Agreement appears to license individual visual displays rather than an overall "total concept and feel." After lengthy consideration, the undersigned has concluded that Judge Schwarzer correctly began his analysis of the issues in the litigation with the 1985 Agreement.[54]

Nevertheless, citing Data East, Judge Walker left open the possibility that "total concept and feel" could still be relevant to the case, but that any evaluation of whether the "total concept and feel" of the works is substantially similar "should occur after unprotectible elements of expression have been identified and excluded from consideration."[55] The court also concluded, citing an earlier Ninth Circuit precedent, Aliotti v. R. Dakin & Co.,[56] that the doctrines of merger and scenes a faire should be applied at the substantial similarity step of the analysis of infringement, and not at the copyrightability step.[57] Accordingly, the court held that "a resolution whether the works in suit are not substantially similar because of the merger of idea and expression in Apple's visual displays is premature at this time."[58]

In a later opinion issued about five months later, Judge Walker again wrestled with the issue of whether "elements of an allegedly infringed work which are found to be 'unprotectible' must be eliminated from consideration in the substantial similarity of expression analysis."[59] Judge Walker recognized that one simply cannot say that if an element is unprotectable standing alone, it automatically can form no part of a substantial similarity comparison, for compilations of unprotectable facts would otherwise never be copyrightable.[60] Similarly, musical compositions could be said to be not copyrightable, for every piece of music is merely a collection of individually unprotectable notes.[61]

---

54 Apple Computer, Inc. v. Microsoft Corp., 759 F. Supp. 1444, 1449 (N.D. Cal. 1991).

55 Id. at 1449 n.8.

56 831 F.2d 898 (9th Cir. 1987).

57 The court acknowledged that other Circuits have applied the merger and scenes a faire doctrine at the copyrightability step, rather than the substantial similarity step, but stated that the court was bound to follow Ninth Circuit precedent. 759 F. Supp. at 1456.

58 Id.

59 Apple Computer, Inc. v. Microsoft Corp., 779 F. Supp. 133, 135 (N.D. Cal. 1991).

60 "Removing unprotectible elements prior to the substantial similarity of expression test would preclude copyright protection for factual compilations containing an innovative selection or arrangement of elements because each element would be eliminated and nothing would be left for purposes of determining substantial similarity." Id.

61 Id.

Judge Walker therefore concluded, consistent with his earlier opinions, that one should not simply remove all elements that may be unprotectable standing alone under various copyright doctrines from any further consideration in the substantial similarity test:

> Some dissection of elements and the application of merger, functionality, scenes a faire, and unoriginality theories are necessary to determine which elements can be used freely by the public in creating new works, so long as those works do not incorporate the same selection or arrangement as that of the plaintiff's work. Because there ought to be copyright protection for an innovative melding of elements from preexisting works, elements which have been deemed "unprotectible" should not be eliminated prior to the substantial similarity of expression analysis. ... Instead, if it is determined that the defendant used the unprotectible elements in an arrangement which is not substantially similar to the plaintiff's work, then no copyright infringement can be found. If, on the other hand, the works are deemed substantially similar, then copyright infringement will be established even though the copyrighted work is composed of unprotectible elements. There is simply no other logical way of protecting an innovative arrangement or "look and feel" of certain works.[62]

In a short opinion issued about eight months later, in April of 1992, Judge Walker amplified how the relationship between unprotectable individual elements and a potentially protectable overall "look and feel" including those elements should be analyzed:

> [C]opyright infringement cannot be predicated solely on the use of ... unprotectible elements from the list of similarities. A look and feel which necessarily follows from the use of these elements likewise cannot be protected. If the Windows 1.0 "look and feel" when supplemented with unprotectible expression leads naturally to the look and feel of the works in question, there is no infringement. If, however, the as yet-unspecified "look and feel" of the Apple works is not the necessary result of the grafting of the unprotectible elements onto the licensed "look and feel" of Windows 1.0, infringement may be shown.[63]

Judge Walker gave no further amplification as to how one is to judge whether an overall look and feel "necessarily follows" from its constituent unprotectable elements. The court did, however, clearly recognize that a whole may be greater than the sum of its unprotectable parts, and, if so, the whole may contain protectable expression.

3. Doctrines that Limit the Scope of Copyright Protection

In August of 1992, after additional submissions by Apple of detailed and comprehensive lists of similarities between the Windows and New Wave user interfaces and the Macintosh user

---

[62] Id. at 135-36.

[63] Apple Computer, Inc. v. Microsoft Corp., 1992 Copyr. L. Dec. ¶26,903, at 25,240 (N.D. Cal. 1992).

interface,[64] Judge Walker issued a very detailed opinion dissecting the Macintosh user interface to determine which particular displays were unprotectable and which were potentially protectable. Again citing Data East, the court noted that such dissection is necessary not only to define the scope of the plaintiff's copyright protection,[65] but for the reason that "[i]f the similarity of the works in suit stems solely from unprotectable features, then the plaintiff's case is missing an essential element of infringement."[66]

The court noted the following limiting factors that must be applied in judging what elements of a computer program user interface are unprotectable:

- Functionality. "Purely functional items or an arrangement of them for functional purposes are wholly beyond the realm of copyright as are other common examples of user interfaces or arrangements of their individual elements – the dials, knobs and remote control devices of a television or VCR, or the buttons and clocks of an oven or stove."[67]

- Standardization. "The similarity of ... functional elements of a user interface or their arrangement in products of like kind does not suggest unlawful copying, but standardization across competing products for functional considerations."[68] The court noted that "overly inclusive copyright protection can produce its own negative effects by inhibiting the adoption of compatible standards."[69]

- Expectations of Users (Scenes a Faire). "Some visual displays are or become so closely tied to the functional purpose of the article that they become standard. If 'market factors play a significant factor in determining the sequence and organization' of a computer program, then those patterns may well be termed ideas beyond the ownership of any one seller."[70] Applying this concept of scenes a faire, the court, in one of the most important holdings of its opinion, ruled that the

---

64 The court insisted that Apple specify in detail the similarities that it alleged formed the basis for its claims of infringement. Accordingly, Apple submitted lists of similarities between the Windows and Macintosh user interfaces, and between the New Wave and Macintosh user interfaces, of approximately 200 items each.

65 Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1021 (N.D. Cal. 1992).

66 Id. at 1020.

67 Id. at 1023.

68 Id.

69 Id. at 1025. Although he noted in dicta the importance of standardization and implied that this factor might limit the scope of copyright protection, Judge Walker did not explicitly rely on this factor in any of the rulings in his various decisions in the case.

70 Id. at 1023 (quoting Plains Cotton Co-Op v. Goodpasture Computer Serv., 807 F.2d 1256, 1262 (5th Cir.), cert. denied, 484 U.S. 821 (1987)).

following five basic features are common to all graphical user interfaces, and are therefore unprotectable:[71]

1. Overlapping windows to display multiple images on a computer screen

2. Iconic representation of familiar objects from the office environment[72]

3. Opening and closing of objects as a means of retrieving, transferring or storing information

4. Menus to store information or functions of the computer in a place that is convenient to reach, but saves screen space for other images

5. Manipulation of icons to convey instructions and to control operation of the computer

4. Dissection of the Macintosh User Interface

The court then proceeded to perform a detailed analytic dissection of the Macintosh user interface, based on the list of similarities Apple had submitted, to determine which of the alleged similarities constituted protectable elements of the Macintosh interface. The court ruled that the vast majority of unlicensed similar elements in the Macintosh user interface were not protectable.[73] The following two tables summarize the court's specific findings as to which features of the Macintosh user interface were unprotectable and which were potentially protectable based on application of the various limiting doctrines noted and other traditional copyright doctrines (such as the idea/expression distinction, the requirement of originality and the words and short phrases doctrine).

TABLE I – ELEMENTS HELD NOT PROTECTABLE

FEATURE	LIMITING DOCTRINE
I. Windows	

[71] 799 F. Supp. at 1026. The court treated these five basic features of a GUI interface as both scenes a faire and ideas.

[72] The court concluded that the "desktop metaphor" underlying the Macintosh user interface – a metaphor suggestive of an office environment with familiar office objects such as file folders, documents and a trash can – was an unprotectible idea. Id.

[73] Id. at 1026-42.

_ The rectangular shape of the windows	Lack of originality
_ The use of a muted background	Lack of originality
_ Overlapping windows per se	Scenes a faire, merger, lack of originality
_ Ability to move a window part on and part off the screen	Functionality (indispensable to the convenient manipulation of information contained in windows)
_ Top-most overlapping window displayed as the active window	Functionality (indispensable to the useful employment of overlapping windows), merger
_ Clicking the mouse to bring a window to the top of a stack of windows	Idea
_ Moving only the outline of a window when it is dragged with the mouse, and moving the window to the new position after the mouse button is released	Functionality (requires less computing power), merger (alternatives are not practical)
_ Redisplaying graphic information of a previously obscured window when an overlying window is moved	Scenes a faire, merger
_ Display of text in windows and dialog boxes in proportionally spaced fonts, rather than monospaced fonts	Idea, merger
_ Use of "muted tones" (black, white and gray) in the interface	Lack of originality
II. Icons	
_ Use of iconic representation for objects	Lack of originality

_ Ability to place windows over icons on the desktop	Idea, functionality (screen space constraints dictate ability to place windows over icons)
_ Placement of an icon's title centered below the icon	Limited choices available
_ Associating a different type of icon with each object type	Idea, lack of originality, scenes a faire
_ Presenting an icon image shaped like a file folder to indicate objects of type "directory"	Lack of originality
_ Two dimensional arrangement of icons within a folder	Idea, scenes a faire
III. Object Opening and Closing	
_ Double clicking on an object, or clicking once and selecting the OPEN command, to open an object	Idea, lack of originality, scenes a faire
_ Dragging an object out of the trash can to reverse the decision to delete it	Idea, no separable expression
_ When an object is opened, the display changes to show the contents of the object, creating the appearance that the icon is transformed into the window	Lack of originality, merger
IV. Menus	
_ "View" menu item that allows a user to view objects as icons or in a tabular list	Idea or process, merger
_ An "Attributes" menu item calling up a dialog box that lists the attributes of a selected object	Idea, lack of originality, merger
_ Words and short phrases in menus, such as "Get Info"	Words and short phrases doctrine

_ Menu items that allow a user to print a listing of a folder's contents, "straighten up" icons along a gridline, create a new folder within an existing folder, supply a default name for a newly created folder, and select every object in a folder at once	Functionality, lack of originality, idea or process, merger
V. Icon Manipulation	
_ Ability to move icons to any position on the screen	Process, indispensable to the idea of manipulating icons by mouse
_ Editing an icon's name by clicking on the name, causing an "I-beam" cursor to appear	Lack of originality
_ Ability to select any combination of one or more icons within a folder window	Merger
_ Presenting a window in the same size and position it had the last time it was open when an icon is opened into a window	Merger
_ Storing position of icons so that at the beginning of each session, icons have the same positions they had at the end of the previous session	Lack of originality, procedure, merger
_ Ability to "drop" an icon into a folder by dragging the icon onto the top of the folder	Scenes a faire
_ Use of reverse video to indicate an icon has been selected	Lack of originality, scenes a faire
_ Use of a special discard folder for deleting objects, and asking a user for confirmation when the user attempts to place an application program in the discard folder	Lack of originality, idea, scenes a faire

TABLE II – ELEMENTS HELD POTENTIALLY PROTECTABLE

I. Windows • Graphics that indicate a particular window is the active window
II. Icons • The appearance of particular icons, if original artwork • Use of a trash can icon to depict the discard function
III. Object Opening and Closing • "Zooming rectangle" animation associated with the opening of an icon into a window or the closing of a window into an icon • "Graying out" an icon that has been opened into a window
IV. Menus • Artwork or unique arrangements of items in a dialog box
V. Icon Manipulation • Visual changes associated with an icon as it is dragged across the screen

Having cataloged the vast majority of the alleged similarities in the Macintosh user interface as "unprotectable," Judge Walker, in a curious and important footnote in his decision, stated the following with respect to such "unprotectable" elements:

> Although the court used the words 'not protectible' to describe its [earlier] rulings ... this was not meant to exclude virtually identical copying from being accountable. Accordingly, in determining whether there is a triable issue of material fact to be decided by 'intrinsic' analysis,[74] the court will consider each feature deemed 'unprotectible' under a virtual identity standard, to the extent there is any separable artistic expression associated with that feature.[75]

[74] The "intrinsic" analysis refers to the second step of the Ninth Circuit's "extrinsic/intrinsic" two-step test of copyright infringement. In the first step, the "ideas" of the works in suit are compared for similarity using an "extrinsic test" or "objective analysis of expression." Analytic dissection of similarities, using expert testimony, is performed. Sid & Marty Krofft Television v. McDonald's Corp., 562 F.2d 1157, 1164 (9th Cir. 1977); Shaw v. Lindheim, 919 F.2d 1353, 1357 (9th Cir.), reh'g en banc denied, 1990 U.S. App. LEXIS 20420 (9th Cir. 1990). In the second step, an "intrinsic test" or "subjective analysis of expression" (the traditional substantial similarity test) is performed. Shaw, 919 F.2d at 1357. The "extrinsic-intrinsic" test was transformed into an "objective-subjective" test by the Shaw case. See Section II.E below.

[75] 799 F. Supp. at 1027 n.19 (emphasis in original).

Elsewhere in the opinion, the court observed that the "law of the Ninth Circuit makes plain ... that when any of the various doctrines that limit the scope of protectibility are in operation, the affected expression may provide a basis for a claim of infringement only if the alleged copy is virtually identical to the plaintiff's version."[76] As Judge Walker would ultimately elaborate in his final opinion in the case, a finding that a user interface element is "unprotectable" does not mean that it can form no part of the analysis of infringement. Rather, Judge Walker used the label "unprotectable" merely to mean that an element is subject to a "limiting" doctrine of copyright law.[77] Such elements are treated differently in the infringement analysis than are elements not subject to a limiting doctrine.

5. A Flow Chart of the District Court's Infringement Test

In Judge Walker's final decision in the case,[78] he sets forth a comprehensive test for the issues that would need to be decided at trial on the ultimate issue of infringement of Apple's copyrights by Microsoft and Hewlett-Packard. This test is complicated and subtle, and is more easily understood when depicted as the flow chart set forth in Figure 1. Although Judge Walker does not articulate his test comprehensively in the final opinion in the step-by-step form depicted in Figure 1, the logic flow of Figure 1 can be inferred from the final opinion, together with some language in the various opinions leading up to the final opinion.

Figure 1 is most easily understood as a depiction of a "physical" flow of elements of the interface into one of two ultimate "buckets" – one in which a substantial similarity test is applied

---

[76] Id. at 1027.

[77] It seems apparent that if an element has been literally copied from another source, and is therefore not original at all to the alleged author, copying of that element by a third party cannot, in and of itself, form a basis for infringement, even if such copying by the third party is "virtually identical." As Judge Walker observed in an earlier opinion, "If a plaintiff *directly copied* the expressive elements of his work from preexisting works, he has no right to preclude others from using those same 'unoriginal' elements." 779 F. Supp. at 134-35 (emphasis added). Judge Walker seems, however, not to have been faced with any elements "directly copied" by Apple from other works.

[78] See Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp. 616 (N.D. Cal. 1993).

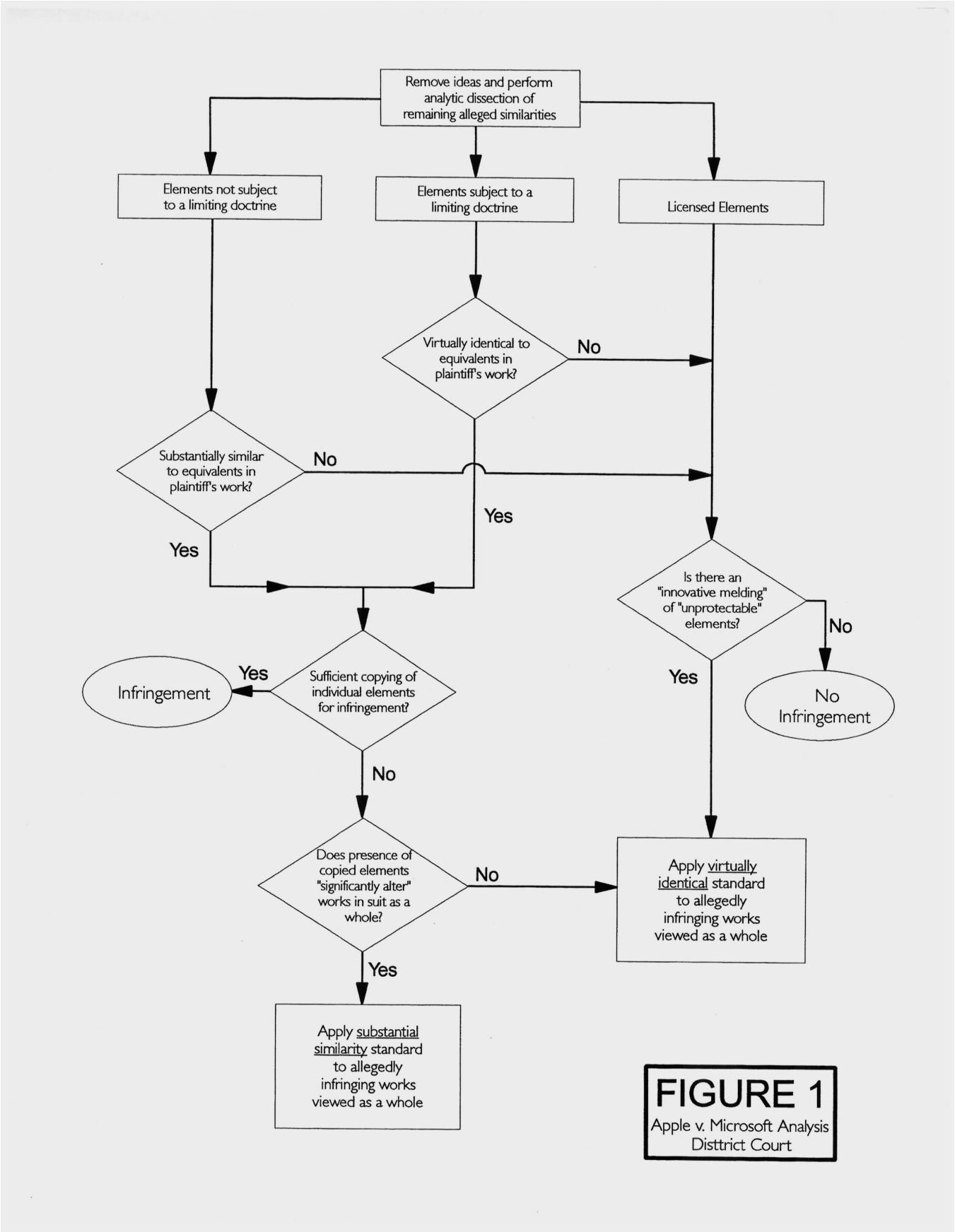


FIGURE 1
Apple v. Microsoft Analysis
Disttrict Court

to those elements comprising the work as a whole, and one in which a virtual identity test is applied to those elements comprising the work as a whole. Along the way, various questions are asked about the elements in order to redirect their flow toward one or the other ultimate "buckets." The steps of the test are explicated below.

(a) Categorization of the Allegedly Similar Elements

The test begins by an analytic dissection of the elements of the defendant's user interface that are allegedly similar to the plaintiff's user interface. Those similarities that are ruled to constitute "ideas" are removed entirely from the subsequent analysis.[79] The remaining similarities are then divided into three categories: (i) elements not subject to a limiting doctrine, (ii) elements subject to a limiting doctrine, and (iii) licensed elements. Although Judge Walker does not explicitly treat the third category of elements separately in his exegesis of his test in the final opinion, the fact that this category of elements was present in the case – and the fact that most alleged similarities in the case fell into this category – determined the course of the analysis from the very beginning of the case. Moreover, from the various opinions, it is clear that Judge Walker treats licensed elements the same as "unprotectable" elements: though individually unprotectable because licensed, they may nevertheless form part of an "innovative melding" of unprotectable elements into some "larger" expression that may (if not also licensed) be infringed. Thus, although many look and feel cases will not raise the issue of licensed elements, they are included for completeness in the flow chart.

Much of Judge Walker's earlier opinions were occupied with this categorization of the allegedly similar elements of the Macintosh interface. The "limiting doctrines" recognized by Judge Walker in his various opinions include the traditional copyright limiting doctrines of lack of originality, scenes a faire, merger, and the words and short phrases doctrine. They also include those doctrines recognized by Judge Walker and discussed previously which are more specific to computer programs: functionality (purely functional items or an arrangement of them for functional purposes), standardization,[80] and expectations of users (visual displays so closely tied to the functional purpose of the article that they become standard).[81]

---

[79] See Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1026 (N.D. Cal. 1992).

[80] See id. at 1023. But see Lotus Development Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 69 (D. Mass. 1990) ("[T]he desire to achieve 'compatibility' or 'standardization' cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual 'work.'").

[81] Several other recent decisions have recognized limiting doctrines or "externalities" that may limit the scope of copyright protection for computer programs. See, e.g., Computer Assocs. v. Altai Inc., 982 F.2d 693 (2d Cir. 1992) (hardware constraints, compatibility, efficiency, computer manufacturers' design standards, widely accepted programming practices, and elements that have entered the public domain through free accessibility); Manufacturers Technologies, Inc. v. CAMS, Inc., 706 F. Supp. 984 (D. Conn. 1989) (hardware constraints). Defendants may be expected to invoke these limiting doctrines in applying the Apple v. Microsoft test in the future. It should be noted, however, that the Second Circuit in the Altai. case explicitly stated that its test was intended for judging infringement of computer program *code* only, and it would not control "categorically distinct works" other than code, such as

The articulation of the test depicted in Figure 1 implicitly assumes that the plaintiff has created a list of allegedly similar elements or features in the defendant's work (or that the court, perhaps with the aid of expert testimony, creates one). As previously noted, it was Apple's original argument that, because Microsoft had created in Version 2.03 a product that was more similar overall to the Macintosh user interface than was Version 1.0, Microsoft had taken itself outside the license from Apple. Apple therefore argued that the court should compare the entire user interface of Windows 2.03 against the entire Macintosh user interface. Apple resisted the court's attempts to force it to create a list of similarities that focused on individual user interface elements.

However, based on Judge Schwarzer's early ruling in the case construing the license agreement between Apple and Microsoft, Judge Walker was of the view that the infringement analysis had to proceed based on individual visual displays of the interface, and he therefore required Apple to produce a list of allegedly similar visual display elements. As discussed in more detail below, it is unclear whether Judge Walker, and other courts in the future, would require the same starting point of a list of individual allegedly similar elements in a case in which a license of individual elements were not involved.

(b) Test for Copying of Individual Elements

Having categorized the allegedly similar elements of the interfaces in question, the first major adjudicative step in Judge Walker's test is to determine whether there has been copying of individual elements. For those elements not subject to a limiting doctrine, a substantially similar test is applied. For those elements subject to a limiting doctrine, a virtually identical test is applied.[82] Judge Walker expresses this step in his final opinion as follows:

---

screen displays. Altai, 982 F.2d at 703. It is arguable, therefore, whether it is appropriate to apply the externalities recognized in Altai for a claim of copying of code to a claim of copying of a user interface.

Moreover, neither Judge Walker nor the Second Circuit in Altai. gives any detailed analysis of how broadly the limiting doctrines or externalities mentioned in the opinions should be construed, nor what criteria should be used in applying them to the facts of any specific case. It is therefore very uncertain at this point what the contours of these doctrines will ultimately be as applied to computer program user interfaces.

[82] Judge Walker relied on Harper House, Inc. v. Thomas Nelson, Inc., 889 F.2d 197 (9th Cir. 1989) for using a virtually identical standard with respect to elements subject to a limiting doctrine. Harper House held that works such as compilations consisting largely of uncopyrightable elements are entitled to copyright protection, but only limited protection. In such cases, infringement may be found only in the event of "bodily appropriation of expression." Id. at 205.

Judge Walker further observed:

> The applicability of the virtual identity standard does not, however, depend on whether the work at issue is deemed to be a compilation, but rather on the range of possible expression. ... Thus, the critical factor in deciding whether to apply a

(1) Applying the appropriate test to each element, are the remaining unadjudicated[83] elements substantially similar or virtually identical to their equivalents in Apple's works?[84]

(i) Where Similarity of Individual Elements is Present. If individual elements in the defendant's work are found substantially similar or virtually identical, as the case may be, to their equivalents in the plaintiff's work, then those elements have been "copied" by the defendant on an individual basis. Before going on to the next major step of the test (adjudication of infringement of the works as a whole), one must ask whether there has been sufficient copying of individual elements to establish infringement even apart from comparison of the works as a whole.

Although Judge Walker does not make this sub-step an explicit part of his final opinion, this sub-step is implicit in an opinion issued by the court on April 14, 1993.[85] In that decision, the court stated that liability could be established by Apple with respect to "unlicensed, artistic expression" associated with certain enumerated items subject to a limiting doctrine (i.e., "unprotectable" in Judge Walker's parlance) if "equivalent expressions" in the defendants' products were "virtually identical" to such items.[86] For items not subject to a limiting doctrine, however, liability could be established by showing that "unlicensed, artistic expression" associated with such items in Apple's products were "substantially similar" to "equivalent expressions" in the defendants' products.[87] Thus, it appears from this language that "virtually identical" copying of any single element or group of individual elements subject to a limiting doctrine, or "substantially similar" copying of any single element or group of individual elements not subject to a limiting doctrine, may potentially form the basis for copyright infringement.

In any event, this sub-step makes logical sense. A defendant could, for example, copy one or more specific icons from a plaintiff's work, even though the defendant might use such icons in the context of a user interface of its own that is not, taken as a whole, substantially similar (or virtually identical) to the plaintiff's user interface. If such icons are sufficiently

---

virtual identity standard is whether the work is capable of only a narrow range of expression, not, as Apple contends, whether the work can be labeled a compilation. Although the works at issue here are not "factual" works *per se*, they are works whose ideas can be expressed in limited ways.

821 F. Supp. at 625.

83 Judge Walker's reference to "unadjudicated elements" basically means allegedly similar elements in Microsoft Windows that were not ruled to have been licensed to Microsoft by Apple.

84 Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp. 616, 622 (N.D. Cal. 1993).

85 See Apple Computer, Inc. v. Microsoft Corp., No. C-88-20149-VRW (N.D. Cal. Apr. 14, 1993).

86 Id. at 3.

87 Id.

fanciful to be protectable individually – and artwork is certainly among the traditional subject matters of copyright protection – then there should be infringement based on such copying of individual elements alone. The scope of damages recoverable for such copying of individual elements might be small, and the plaintiff may still want to establish infringement under the next step of the test in its work taken as a whole. Nevertheless, for purposes of an adjudication of infringement, one must ask whether there has been sufficient copying of individual elements alone.[88]

(ii) Where Similarity of Individual Elements is Not Present. If individual elements are found not to be substantially similar or virtually identical, as the case may be, to their equivalents in the plaintiff's work, then Judge Walker treats those elements the same as if they had been licensed. Specifically, such non-similar elements, together with the licensed elements, may form part of some "larger" expression in the plaintiff's work that is more than the sum of its parts. One must therefore group together such non-similar and licensed elements in order to determine, in the next step of the analysis, whether such "larger" expression exists, and if so, whether it has been infringed.

(c) Test for Infringement of the Works as a Whole

(i) The Test Where There Has Been Copying of Individual Elements. The second and third major adjudicative steps of Judge Walker's test judge whether there has been infringement of the works in suit viewed as a whole. Judge Walker expresses these steps as follows in his final opinion:

> (2) If [the answer to the question in step (1) is yes], does the presence of the elements found to be substantially similar or virtually identical significantly alter the works in suit as a whole?
>
> (3) If the answer to (2) is yes, are the allegedly infringing works, viewed as a whole, substantially similar to the copyrighted work?
> If the answer to (2) is no, are the allegedly infringing works, viewed as a whole, virtually identical to the copyrighted work?[89]

These steps, which apply only where copying of individual elements has been found under the first adjudicative step, are reflected in the lower half of Figure 1. Specifically, those individual elements that have been found "copied" in step (1) (under either a substantially similar or virtually identical standard) are grouped together on the left side of the lower half of Figure 1, and one asks whether those elements "significantly alter" the works in suit as a whole. If the answer is yes, then a substantial similarity standard is applied to the allegedly infringing works

---

[88] Judge Walker recognizes in his final opinion the long-standing doctrine of *de minimis* copying, in which the amount copied is so small as to not justify a finding of infringement. See 821 F. Supp. at 623-24. Although raised in the context of adjudicating similarity of the works as a whole in the final step of his test, Judge Walker would presumably apply the doctrine to copying of individual elements as well.

[89] Id. at 622-23.

viewed as a whole. If the answer is no, then a virtually identical standard is applied to the allegedly infringing works viewed as a whole.

Judge Walker nowhere in any of his opinions in the case explicates precisely what he means by asking whether these elements that have been found individually similar "significantly alter" the works in suit as a whole. It seems apparent, however, that by these steps of his test, Judge Walker is seeking to determine whether there is some "larger" expression present in the plaintiff's work that is greater than the sum of its individual parts, and which expression one might lose sight of in the course of the focus solely on individual elements in the previous portions of the test. If there is some "larger" expression present that is more than the sum of its individual parts, then the works are compared as a whole under a substantially similar test. If there is no "larger" expression present – i.e., the elements found similar simply are what they are individually in the interface and do not form part of some "larger" expression in the interface – then the works are compared as a whole under a virtually identical standard.

An analogy may help illuminate the goals of these steps of the test. Consider how one might make a comparison of two puzzles made up of a collection of many individual pieces. One would first look for similarity among the individual pieces of the respective puzzles to determine which ones were similar. One would then fit the similar pieces of the plaintiff's puzzle together to see if a larger picture forms, such as the Bavarian Alps. If so, one would then fit the corresponding pieces of the defendant's puzzle together to see if the Bavarian Alps appear (or at least a mountain scene of some kind).[90] If the Bavarian Alps (or at least a mountain scene)

---

[90] It is not entirely clear from Judge Walker's articulation of the second and third adjudicative steps of his test whether he intends one to focus on the plaintiff's work or the defendant's work (or both) in asking whether the copied elements "significantly alter" the works in question. At one point in his opinion, Judge Walker analyzes whether a particular copied element – a trash can icon – significantly alters the defendant's work (New Wave), and concludes that it does not because it is only one of many such icons displayed on the default screen of the defendant's work. Id. at 624.

One could argue that this focus on the defendant's work is wrong-headed, because the trash can icon might not significantly alter the defendant's work if the defendant had added other icons on screen with it, whereas such trash can icon might significantly alter the plaintiff's work if it were the only icon on the default screen. Focusing on the defendant's work, then, risks allowing elements not copied by the defendant to affect the outcome of the question whether the copied element significantly alters the work. This seems contrary to the well established principle in copyright law that the defendant may not avoid a claim of infringement by showing how much protectable expression and/or other elements the defendant's work contains that are not similar to the plaintiff's work. See, e.g., Computer Assocs. v. Altai, Inc., 982 F.2d 693, 714 (2d Cir. 1992).

In the next paragraph of the opinion, however, Judge Walker focuses on whether "four isolated similarities" are "qualitatively significant to Apple's works as a whole." 821 F. Supp. at 624. Judge Walker therefore appears to be asking the question whether copied elements significantly alter the works as a whole with respect to both the plaintiff's and the defendant's works.

appear, then as the last step of the test, one would compare the two pictures of the mountains to see if they were substantially similar. If a mountain scene did not appear when the puzzle pieces were fit together, then as the last step of the test one would simply compare the collection of similar puzzle pieces in each of the works to see if they were virtually identical.

Judge Walker determined that this branch of the flow chart depicted in Figure 1 applied to the alleged similarities in Hewlett-Packard's New Wave product. In particular, Judge Walker found that a reasonable jury could find that some of the allegedly similar individual (and unlicensed) elements in the New Wave interface were copied from Apple's works. He determined, however, that as a matter of law such elements, even if found to have been "copied" individually, did not significantly alter the works in suit as a whole. Accordingly, he ruled that the virtually identical standard must be applied when judging Apple's and Hewlett-Packard's works viewed as a whole for similarity.[91]

(ii) The Test Where There Has Not Been Copying of Individual Elements. Where copying of individual elements has not been found under the first adjudicative step, the test for infringement of the works in suit viewed as a whole is reflected in the right hand side of the lower half of Figure 1. Specifically, those individual elements that have been found not "copied" in step (1) are grouped together with licensed elements on the right side of Figure 1, and one asks whether there is an "innovative melding" of those elements. If the answer to this question is yes, then a virtually identical standard is applied to the allegedly infringing works viewed as a whole. If the answer is no, then there is no infringement.

Judge Walker ultimately determined that the alleged similarities in Microsoft's Windows product fell into this branch of the flow chart depicted in Figure 1. Specifically, Judge Walker ruled that as a matter of law, "Microsoft's Windows consists only of elements that are either unprotectable, licensed, or protectable but lacking sufficient similarity to Apple's works."[92] He went on to note, however, that even so, there might still be infringement of Apple's copyrights:

> As noted by the court in Apple Computer, Inc. v. Microsoft Corp., 779 F. Supp. 133, 135-36 (N.D. Cal. 1991), copyright protection may attach to an innovative "melding" of unprotectable features. ... [B]ecause the court has determined that Windows consists only of elements that are either unprotectable, licensed, or protectable but lacking sufficient similarity to

---

If the real goal of the "significantly alter" test is to look for "larger" expression that may have been copied beyond individual elements, then it makes sense to ask whether the copied elements significantly alter the works as a whole with respect to both the plaintiff's and the defendant's works. Returning to the analogy set forth in the text, one must fit the pieces together of both puzzles to determine whether they both contain the Bavarian Alps (or at least a mountain scene of some kind). Only if both puzzles contain the larger picture will there be any "larger" comparison to be made beyond the individual pieces of the respective puzzles.

91 821 F. Supp. at 623.

92 Id.

> Apple's works, the jury must apply the virtual identity standard when comparing the similarities of Microsoft's works as a whole to Apple's.[93]

Judge Walker does not articulate in his various opinions precisely what he means by asking whether there is an "innovative melding" of unprotectable elements. It seems likely, however, that this question is aimed at determining much the same issue as the question whether the presence of copied elements "significantly alters" the works in suit in the left half of Figure 1 – namely, do the unprotectable elements at issue form some "larger" expression in the plaintiff's work that is greater than the sum of its individual, unprotectable, parts? If so, and if such "larger" expression has been copied virtually identically, then there is infringement.

(iii) The Outcome of the Case. Judge Walker ultimately determined that a virtually identical standard of similarity should be applied to both Microsoft's Windows and Hewlett-Packard's New Wave products.[94] In the case of Windows, he traversed the right hand side of Figure 1 and ended up in the virtually identical "bucket" because he found all allegedly similar elements in the interface to be either licensed, unprotectable, or protectable but not "copied." In the case of New Wave, he traversed the left hand side of Figure 1 and ended up in the virtually identical "bucket" because, although he found some such similarities potentially protectable and potentially copied, he found that they did not, as a matter of law, significantly alter the works in suit as a whole.

Following Judge Walker's rulings in the final opinion, Apple stipulated with Microsoft and Hewlett-Packard that it would not oppose motions for summary judgment in favor of Microsoft and Hewlett-Packard on the ground that Windows and New Wave are not virtually identical in user interface to Apple's Macintosh operating system.[95] Based on such stipulation, Judge Walker entered summary judgment for Microsoft and Hewlett-Packard, placing the case in a posture for an appeal to the Ninth Circuit by Apple.

THE NINTH'S CIRCUIT'S DECISION

On appeal to the Ninth Circuit, Apple mounted two fundamental challenges to the district court's decisions.[96] *First*, Apple argued that the district court misconstrued the license between Apple and Microsoft, and that the court should not have allowed the license for Windows 1.0 to serve as a partial defense to Microsoft's alleged copying in Windows 2.03 and 3.0. *Second*, Apple argued that the district court should not have dissected Apple's works, but rather should have allowed the jury to perform an overall comparison of the works in suit (including both licensed and unlicensed elements) under a traditional standard of substantial similarity, rather than a standard of virtual identity.

---

93 Id.

94 Id. at 625.

95 See Statement of Non-Opposition, Apple Computer, Inc. v. Microsoft Corp., No. C 88 20149 VFW (N.D. Cal. May 26, 1993).

96 Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1438-39 (9th Cir. 1994), cert. denied, 115 S. Ct. 1176 (1995).

1. The Interpretation of the License Between Apple and Microsoft

With respect to Apple's first challenge on appeal, the Ninth Circuit affirmed the district court's interpretation of the license between Apple and Microsoft to be a license of individual "visual displays," not of the Windows 1.0 interface as a whole.[97] The Ninth Circuit agreed with the district court that the license meant that Microsoft was free to use any visual display appearing in the Windows 1.0 interface in any combination or arrangement in any future product, even if doing so would make "the interface as a whole look more 'Mac-like' than Windows 1.0"[98]

The Ninth Circuit rejected Apple's argument that Microsoft could not rely on the license as a defense, based on a line of cases holding that if a licensee exceeds the scope of a license, the license may not be relied upon as a defense to copyright infringement.[99] The court distinguished the cases cited by Apple on the ground that "[t]he cases on which Apple relies ... merely establish that the breach of a *prohibition* in the license agreement can lead to a finding of infringement."[100] The court concluded that "[w]here, as here, the accused works include both licensed and unlicensed features, infringement will depend on whether the unlicensed features are entitled to protection."[101]

As in the case of the district court's analysis, the Ninth Circuit recognized that its construction of the license agreement to allow the use of individual visual displays appearing in Windows 1.0 would be pivotal to the analysis of the case: "The fact that Apple *licensed* the right to copy almost all of its visual displays fundamentally affects the outcome of its infringement claims."[102] Moreover, as described below, the court's construction of the license caused it, like the district court, to focus the analysis of infringement on individual screen elements.

2. The Test for Infringement

(a) Some Preliminary Observations

With respect to Apple's challenge of the district court's analysis of the alleged similarities in the works at issue, the Ninth Circuit ruled that the "district court's approach was

---

97 Id. at 1439.

98 Id. at 1442.

99 Id. at 1441-42. See, e.g., S.O.S., Inc. v. Payday, Inc., 886 F.2d 1081, 1087 (9th Cir. 1989) ("A licensee infringes the owner's copyright if its use exceeds the scope of its license.").

100 Apple v. Microsoft, 35 F.3d at 1441 (emphasis added). For example, the Ninth Circuit noted that in the S.O.S. case, the license granted only the right to *use* a copyrighted computer program; the licensee exceeded the scope of the license by preparing a modified version of the program without the licensor's permission.

101 Id.

102 Id. at 1442 (emphasis in original).

on target."[103] Before turning to a detailed analysis of the Ninth Circuit's opinion, however, it is important to note for purposes of construing some of the ambiguities and unstated assumptions in that opinion, that the Ninth Circuit chose to use the district court's analysis as its starting point, rather than to start from the "ground up" in setting forth a test for infringement.

For example, the Ninth Circuit explicitly noted that the "district court ... had to cut new paths as it went along; we have the luxury of looking at the case at the end of the trip."[104] The court further noted that the district court had been faced with a case having "the unusual, added complexity of a license that arguably covers some or most of the allegedly infringing works."[105] Accordingly, the Ninth Circuit was willing to simply review the "new paths" that the district court had cut, and found that the district court's analysis followed "naturally from a long line of copyright decisions which recognizes that works cannot be substantially similar where analytic dissection demonstrates that similarities in expression are either authorized, or arise from the use of common ideas or their logical extensions."[106]

Because the Ninth Circuit found that the district court's approach to its infringement analysis was basically "on target," the analysis and flow chart set forth below assume that the Ninth Circuit did not disagree with the district court's analysis with respect to certain issues that appear in the Ninth Circuit's opinion – sometimes only in passing reference – but which are not as fully fleshed out as in the district court's various opinions. There are, however, some issues with respect to which the Ninth Circuit's opinion speaks in somewhat different terms than those used by the district court, and there appear to be some other nuances of analytical detail in which the district court's and the Ninth Circuit's opinions differ. Such differences are analyzed and embodied in the accompanying flow chart and discussion.

(b) The Ninth Circuit's Three Step Test

The Ninth Circuit articulated the following three step test for copyright infringement:

1. The plaintiff must identify the *source(s)* of the alleged similarity between his work and the defendant's work.

2. Using analytic dissection, and, if necessary, expert testimony, the court must determine whether any of the allegedly similar features are protected by copyright. Where, as in this case, a license agreement is involved, the court must also determine which features the defendant was authorized to copy. Once the scope of the license is determined, unprotectable ideas must be separated from potentially protectable expression; to that expression, the court must then apply the relevant limiting doctrines in the context of the particular medium involved, through the eyes of the ordinary consumer of that

---

[103] Id. at 1439.

[104] Id.

[105] Id.

[106] Id.

product.[107]

3. Having dissected the alleged similarities and considered the range of possible expression, the court must define the scope of the plaintiff's copyright – that is, decide whether the work is entitled to "broad" or "thin" protection. Depending on the degree of protection, the court must set the appropriate standard for a subjective comparison of the works to determine whether, as a whole, they are sufficiently similar to support a finding of illicit copying. [108]

(c) A Flow Chart of the Test

As depicted in the flow chart of Figure 2, the Ninth Circuit's test is comprised of three substantive components:

- Identification of the alleged similarities between the plaintiff's and the defendant's works;
- Analytic dissection of the alleged similarities; and
- Comparison of the alleged similarities at two levels: individual elements and the works as a whole.

[107] It is unclear why the Ninth Circuit is applying an "ordinary consumer" standard in the analytic dissection step, particularly when the court has noted that expert testimony is appropriate to assist the court in performing its dissection, and in view of the fact that the rulings to be made in the dissection step as to which alleged similarities are protectable and which are not are primarily rulings of law. The court nowhere in its opinion articulates what its reference to the "ordinary consumer" means, nor how it affects the analysis.

[108] 35 F.3d at 1443 (emphasis in the original).

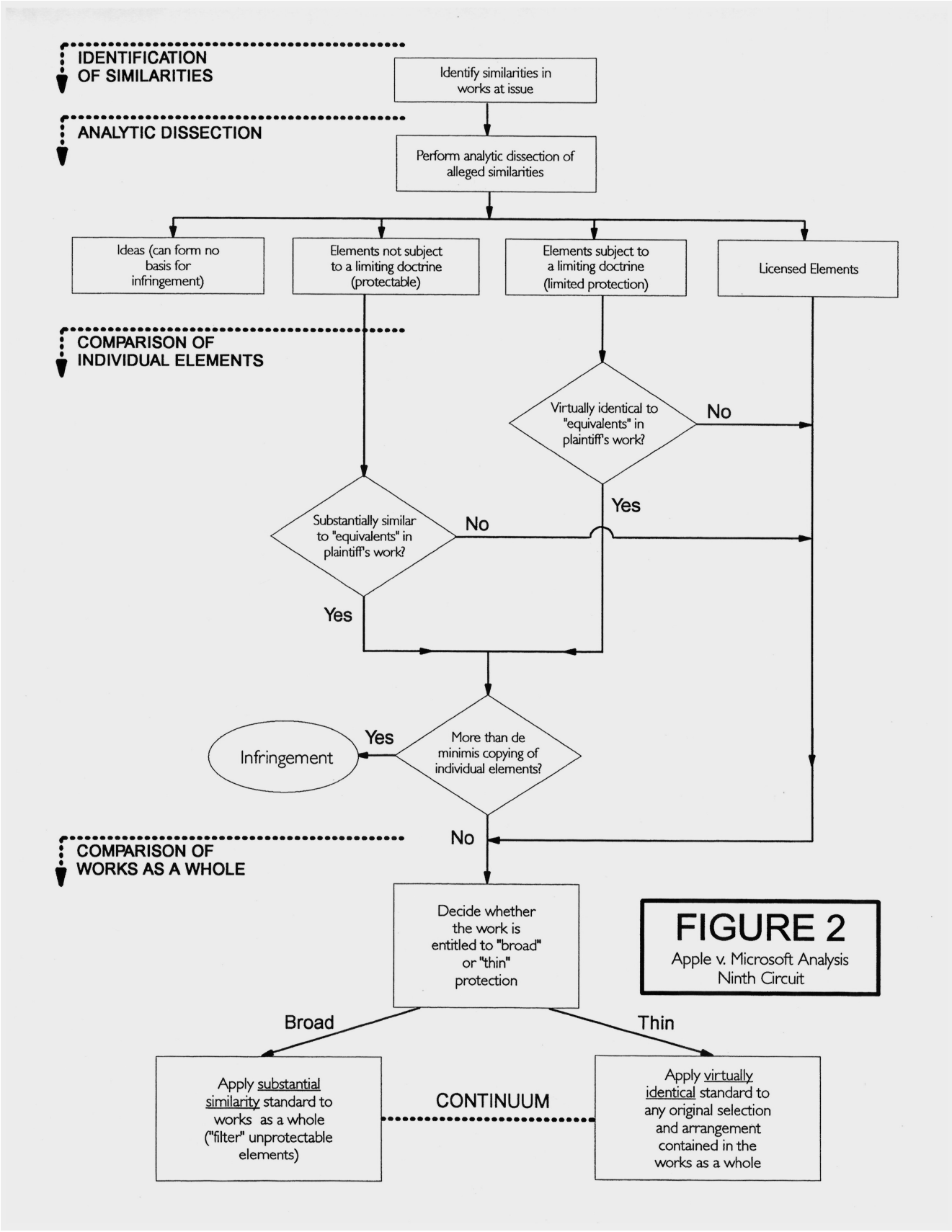


FIGURE 2
Apple v. Microsoft Analysis
Ninth Circuit

(i) Identification of Similarities. The first component of the Ninth Circuit's infringement test is an identification of the alleged similarities between the plaintiff's and the defendant's works. The district court had required Apple to provide a list of alleged similarities. Apple strongly resisted doing so on the ground that its interface should not be dissected into component parts and similarities, but rather should be viewed as an integrated work for comparison purposes. Apple raised the issue again on appeal, but the Ninth Circuit rejected Apple's arguments:

> Apple's suggestion that its arm was twisted to provide this list of similarities and that it was somehow inappropriate for the district court to ask for a list and to rely on it, instead of considering the works as a whole, is misplaced. The court had the benefit of numerous videotapes and demonstrations of the GUIs "as a whole." The district court was nevertheless obliged to identify similarities, determine their source, and decide which elements are protectable. It was thus well within the court's case management discretion to ask for a list from Apple.[109]

Two observations may be made with respect to this issue. *First*, the Ninth Circuit does not state whether it is mandatory that a plaintiff provide a list of similarities in every case. The court specifically noted that it was "within the court's case management discretion" to ask for such a list. It appears that if the plaintiff were not to supply this list as part of its pleadings, the court could either require that the plaintiff do so or create such a list as part of its own analysis of what elements of the work are protectable in the dissection step.

Second, the Ninth Circuit does not address at what level or levels of abstraction the similarities on such a list should be defined. As previously noted in the discussion concerning the district court's opinions, Judge Walker's analysis focused mostly on similarities at the "widget" level,[110] perhaps in large part because most of the alleged similarities submitted by Apple on its list were at this level, and because the court had ruled that the license between Apple and Microsoft permitted Microsoft to use widget elements in any combination in current or future products. Apple's list also included similarities at higher levels of abstraction, such as the desktop metaphor and overlapping windows, but Judge Walker ruled that most of such higher level similarities were unprotectable ideas.

Thus, the Ninth Circuit's opinion simply does not spell out at what level of abstraction a list of similarities claimed by a plaintiff as a basis for infringement should be defined. Presumably a plaintiff could focus at whatever one or more levels it felt appropriate to its particular case. A focus on only similarities at high levels of abstraction would apparently not, however, preclude the court from dissecting such similarities at lower levels of abstraction in the next component of the test.

---

[109] Id.

[110] The lowest level of individual graphical user interface components are often called "widgets." Examples of widgets are icons, scroll bars, sliders, buttons, and close boxes. Widgets are used to build up larger components of a graphical user interface, such as dialog boxes and windows.

(ii) Analytic Dissection of the Alleged Similarities. The second component of the Ninth Circuit's test is analytic dissection of the alleged similarities. Expert testimony may be used to assist the court in performing such dissection. Its purpose is to "determine whether any of the allegedly similar features are protected by copyright."[111] Under the Ninth Circuit's analysis, analytic dissection encompasses two tasks:

- Identification and removal of ideas; and
- Application of the limiting doctrines of copyright law.[112]

As depicted in Figure 2, analytic dissection is performed to separate the similarities on the plaintiff's list into ideas, expressive elements of similarity not subject to a limiting doctrine, expressive elements of similarity subject to a limiting doctrine, and licensed elements. Ideas can form no basis for infringement, and must therefore be entirely removed in the dissection step from further consideration. The Ninth Circuit's opinion contains no discussion concerning how one separates ideas from expression, but presumably it is by application of the standard "abstractions test."[113] The Ninth Circuit affirmed the district court's ruling that the five basic components of the desktop metaphor (overlapping windows, iconic representation of objects, manipulation of icons to effect instructions and controls, use of menus, and opening and closing of objects to retrieve, transfer and store information) constitute unprotectable ideas.[114]

Following the identification and removal of ideas from the analysis, the court then applies the various copyright limiting doctrines to the remaining similarities to determine which are protectable. Although Apple did not challenge on appeal the district court's many applications

---

[111] 35 F.3d at 1143. The court further noted, "Dissection is not inappropriate even though GUIs are thought of as the 'look and feel' of a computer, because copyright protection extends only to protectable elements of expression." Id. at 1439.

[112] "Once the scope of the license is determined, unprotectable ideas must be separated from potentially protectable expression; to that expression, the court must then apply the relevant limiting doctrines in the context of the particular medium involved, through the eyes of the ordinary consumer of that product." Id. at 1443.

[113] If an abstractions test is used, then the Ninth Circuit's test has a similar high-level structure to the three-step "abstraction/filtration/comparison" test first articulated by the Second Circuit in Computer Associates Int'l v. Altai Inc., 982 F.2d 693 (2d Cir. 1992), discussed below, and later adopted by the Tenth Circuit in Gates Rubber Co. v. Bando American, Inc., 9 F.3d 823 (10th Cir. 1993) and by the Fifth Circuit in Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994), supplemental opinion & reh'g en banc denied, 46 F.3d 408 (5th Cir. 1995), both of which are discussed below. The Ninth Circuit's analytic dissection step corresponds to the "filtration" step. The Ninth Circuit noted the similarity of its own test to that of the Altai case. 35 F.3d at 1445.

[114] See 35 F.3d at 1443-44.

of the limiting doctrines to individual alleged similarities,[115] the Ninth Circuit noted the following limiting doctrines and their applicability to the case at hand:

- Merger: The court held that an iconic image shaped like a page is merged with the idea of a document stored in a computer program.[116]

- Scenes a Faire: The court held that overlapping windows per se are scenes a faire, although it noted that Apple's particular expression of them may be protectable.[117]

- Functionality: The court ruled that "the ability to move icons to any part of the screen exemplifies an essentially functional process" that was not protectable.[118]

- Hardware Constraints: "[H]ardware constraints limit the number of ways to depict visually the movement of a window on the screen; because many computers do not have enough power to show the entire contents of the window as it is being moved, the illusion of movement must be shown by using the outline of a window or some similar feature."[119]

- Environmental and Ergonomic Factors: "Design alternatives are further limited by the GUI's purpose of making interaction between the user and the computer more 'user-friendly.' These, and similar environmental and ergonomic factors which limit the range of possible expression in GUIs, properly inform the scope of copyright protection."[120]

- Originality: "As the Supreme Court recently made clear, protection extends only to those components of a work that are original to the author, although original selection and arrangement of otherwise uncopyrightable components may be protectable."[121]

---

[115] "Although it does not concede that limiting doctrines were correctly applied to each alleged similarity, Apple does not ask us to review the many discrete decisions reflected in the district court's published opinions. We have done so only to the extent of being satisfied that none makes a difference to the outcome, because we agree that the appeal turns on whether the district court's *approach* was correct." Id. at 1438 n.5 (emphasis in original).

[116] Id. at 1444.

[117] Id.

[118] Id.

[119] Id. at 1445.

[120] Id.

[121] Id. Apple admitted that it had "borrowed heavily from the iconic treatments in the Xerox Star and an IBM Pictureworld research report," id., but disputed several of the district court's individual determinations of originality, claiming for example that its file folder and page

(iii) Comparison of the Alleged Similarities. The final step of the Ninth Circuit's test is comparison of the alleged similarities that remain after the dissection step. As depicted in Figure 2, such comparison must apparently be made at two levels: individual elements and the works as a whole.

(1) Comparison of Individual Elements. With respect to individual elements, the Ninth Circuit's opinion does not explicitly treat comparison of individual elements separately from comparison of the works as a whole. One may infer on two grounds, however, that the Ninth Circuit at least implicitly agreed with Judge Walker that infringement may be found based on sufficient copying of individual elements. *First*, the Ninth Circuit stated generally that it found the district court's approach to be "on target."[122]

Second, the Ninth Circuit's opinion explicitly notes – without any language of disapproval – that the district court ruled that comparisons were to be made at both an individual level and of the works as a whole: "The [district] court then held that those [protectable] elements in NewWave would be compared with their *equivalent* Apple elements for substantial similarity, and that the NewWave and Windows 2.03 and 3.0 works as a whole would be compared with Apple's works for virtual identity."[123] The opinion also contains a footnote suggesting that it recognized that infringement could potentially be based on sufficient copying of individual elements (i.e., more than trivial copying under the *de minimis* doctrine as depicted in Figure 2):[124]

> The court's order that the four individual similarities in NewWave were to be compared at trial with their 'equivalents' in Apple's works for substantial similarity, Apple VI, 821 F. Supp. at 631, is not an issue on appeal. Apple does not assert infringement as to any of these elements individually, and we therefore assume that it did not oppose entry of judgment on this basis. In any event, as the district court held, id. at 623-25, these similarities do not comprise a core of protectable and unlicensed similarities substantial enough to warrant a finding of illicit copying under a standard of substantial similarity.

The Ninth Circuit does not address the issue of the standard to be applied – substantial similarity or virtual identity – to adjudicate copying of individual elements. It seems likely, however, that the court would determine that standard based upon whether or not the elements at issue were subject to a limiting doctrine. Such was the approach used by the district court. Moreover, the Ninth Circuit adopted a similar approach, as discussed in the next subsection, in determining the standard to be applied in comparing the works as a whole. Accordingly, Figure

---

icon designs were original. The Ninth Circuit responded with an apparent invocation of the *de minimis* doctrine: "Even if they are, these particular icons add so little to the mix of protectable material that the outcome could not reasonably be affected." Id.

[122] Id. at 1439.

[123] Id. at 1438 (emphasis added).

[124] Id. at 1439 n.6.

2 depicts the standard to be applied for individual copying as turning on whether or not the element at issue is subject to a limiting doctrine. If not, a standard of substantial similarity is used. If so, a standard of virtual identity is used.

(2) Comparison of the Works as a Whole. The final step in the Ninth Circuit's test is a comparison of the works as a whole. Because of Apple's stipulation that it would not oppose motions for summary judgment that the defendants' works were not virtually identical as a whole, neither the district court nor the Ninth Circuit was called upon to spell out precisely what role the elements that have been determined to be either unprotectable or licensed play in comparing the works as a whole.

It is clear from the Ninth Circuit's opinion that unprotectable and licensed elements play a role in determining what standard of similarity is to be applied to the comparison of the works as a whole: "When the range of protectable and unauthorized expression is narrow, the appropriate standard for illicit copying is virtual identity."[125] The Ninth Circuit held that because the district court had found *no* unlicensed protectable similarities in Windows, and had found only a handful in New Wave, only Apple's "unique selection and arrangement" of these features could form the basis for infringement, and the Ninth Circuit affirmed the district court's adoption of a standard of virtual identity for the comparison of similarity.[126]

Interestingly, the Ninth Circuit recognized a continuum between affording "thin" protection under a virtual identity standard to works for which the alleged similarities relate mostly to unprotectable elements, and affording "broad" protection under a substantial similarity standard to works for which the alleged similarities relate mostly to protectable elements. "Which end of the continuum a particular work falls on is a call that must be made case by case. We are satisfied that this case is closer to Frybarger[127] [holding that the similarities between the works were confined to ideas and general concepts] than to McCulloch[128] [holding that an artistic work like a decorative plate receives broader protection because of endless variations of expression available to the artist]."[129] The court does not state how one is to locate a particular case on that continuum, nor what standard of similarity is to govern at various points along the continuum between the extremes of virtual identity and substantial similarity.

It is unclear from the Ninth Circuit's opinion how unprotectable elements are to be treated in the actual comparison of the works as a whole. In particular, in the case of a jury trial, are such elements always or sometimes excluded entirely from the jury, or are the works as a whole presented to the jury – including the unprotectable elements – and the jury instructed to ignore the unprotectable elements in comparing the works? If there is protectable expression in the selection and arrangement of unprotectable elements in the plaintiff's work, how is the jury

---

[125] Id. at 1439.

[126] Id. at 1446.

[127] Frybarger v. International Business Machines Corp., 812 F.2d 525 (9th Cir. 1987).

[128] McCulloch v. Albert E. Price, Inc., 823 F.2d 316 (9th Cir. 1987).

[129] Apple v. Microsoft, 35 F.3d at 1447.

instructed to consider only such selection and arrangement as distinct from the individual unprotectable elements that comprise it, and how is the protectable selection and arrangement identified to the jury?

The Ninth Circuit's opinion contains unclear language about how unprotectable elements are to be treated in the comparison of the works as a whole. One the one hand, the court stated that "the party claiming infringement may place '*no* reliance upon any similarity in expression resulting from' unprotectable elements. ... Otherwise, there would be no point to the extrinsic test, or to distinguishing ideas from expression."[130] If *no* reliance may be placed upon unprotectable elements, then perhaps such elements should be excluded from the jury.

On the other hand, the court continued, "This does not mean that at the end of the day, when the works are considered under the intrinsic test, they should not be compared as a whole. ... Nor does it mean that infringement cannot be based on original selection and arrangement of unprotected elements. However, the unprotectable elements have to be identified, or filtered, before the works can be considered as a whole."[131] The court's reference to selection and arrangement suggests that in some instances at least individually unprotectable elements will have to be presented to the jury to adjudicate whether protectable selection and arrangement have been copied. However, the court's statement that unprotectable elements must be "filtered ... before the works can be considered as a whole" suggests that in other instances it may be appropriate not to allow unprotectable elements to be presented to the jury.

Because of Apple's stipulation that resulted in skipping a jury trial in favor of an immediate appeal with respect to the district court's legal rulings concerning the applicable standard of similarity, no jury instructions were ever written or presented to either the district court or the Ninth Circuit. The Ninth Circuit's opinion contains only two oblique references to the issue of how unprotectable elements are to be treated before a jury. *First*, the court noted that "Apple also argues that even if dissection were appropriate, the district court should not have eliminated from jury consideration those elements that are either licensed or unprotected by copyright."[132] Because no jury instructions were ever at issue in the district court proceedings, however, it is unclear what Apple's basis was for arguing that the district court "eliminated from jury consideration" the unprotectable and licensed elements. In any event, the Ninth Circuit recharacterized Apple's challenge as merely an alternative formulation of the following fundamental contention on appeal: "Apple wants an *overall comparison* of its works to the accused works for *substantial similarity* rather than virtual identity."[133]

Second, the Ninth Circuit noted that Harper House, Inc. v. Thomas Nelson, Inc.[134] reversed a district court's ruling "because 'total impact and effect' test of [the] jury instruction

---

[130] Id. at 1446 (quoting Aliotti v. R. Dakin & Co., 831 F.2d 898, 901 (9th Cir. 1987)) (emphasis added by the court).

[131] Apple v. Microsoft, 35 F.3d at 1446.

[132] Id. at 1442.

[133] Id. (emphasis in original).

[134] 889 F.2d 197 (9th Cir. 1989).

did not distinguish between protectable and unprotectable material, thereby improperly making it possible for [the] jury to find copying based on unprotected material instead of selection and arrangement."[135] Although this language does not specify when unprotectable elements may properly be presented to a jury, it makes clear that if such material is presented to the jury, the jury must be instructed as to what is, and is not, protectable in the plaintiff's work.

It therefore appears that the questions of whether and how one presents unprotectable elements to a jury in a "look and feel" case, and what form the accompanying jury instructions must take, remain open at this time and must await resolution in some other case.

(iv) The Differences in the Comparison Step Between the District Court and the Ninth Circuit. It is apparent from a comparison of Figure 1 and Figure 2 that there are some differences between the district court's and the Ninth Circuit's articulation of the comparison step and the associated standard of similarity that is to be applied. As depicted in Figure 1, the district court's test made the choice of standard depend upon whether individually copied elements "significantly alter" the works in suit as a whole, and whether there was an "innovative melding" of individually un-copied but unprotectable elements. If there was no "innovative melding" of un-copied elements, there could be no infringement. If there was an "innovative melding," a standard of virtual identity would apply. Similarly, if the presence of copied elements "significantly altered" the work as a whole, a standard of substantial similarity would apply. Otherwise, a standard of virtual identity would apply.

The Ninth Circuit's opinion does not speak in terms of an "innovative melding" or a "significant alteration" of the work as a whole, nor does its test separate copied elements from un-copied elements in choosing the standard of similarity to be applied. Rather, as previously discussed, the Ninth Circuit's test chooses the standard of similarity based on the range of expression possible in the protectable elements and whether the alleged similarities are comprised mostly of unprotectable elements. The Ninth Circuit's test sets up a continuum between (i) works that are comprised mostly of protectable elements for which the range of expression is not narrow – which are to be afforded "broad" protection under a substantial similarity standard, and (ii) works that are comprised wholly or mostly of unprotectable elements and for which protectable expression resides wholly or mostly in the original selection and arrangement of unprotectable elements – which are to be afforded "thin" protection under a virtually identical standard.

(d) Outcome of the Case

The Ninth Circuit concluded that, because the district court found that the similarities in Windows consisted only of unprotectable or licensed elements, and that the similarities between protectable elements in Apple's works and New Wave were *de minimis*, the district court properly concluded that a standard of virtual identity must govern a comparison of the works as a

[135] Apple v. Microsoft, 35 F.3d at 1446.

whole. "Accordingly, since Apple did not contest summary judgment under the virtual identity standard on the merits, judgment was properly entered."[136]

SOME OBSERVATIONS ABOUT THE TESTS ADOPTED BY THE DISTRICT COURT AND THE NINTH CIRCUIT

It is apparent from the flow charts of Figure 1 and Figure 2 that both Judge Walker and the Ninth Circuit have formulated complex tests composed of many individual steps that may prove very difficult to apply in future cases. In spite of the complexities and complications, it seems they were attempting through their tests to avoid overprotection of a copyrighted work, while at the same time trying not to lose sight of expression that might be present in a work consisting largely of licensed and/or unprotectable elements. For example, in an unpublished order issued about a month before his final opinion, Judge Walker wrote:

> Common sense supports [the application of a virtual identity standard]. The fact that the work as a whole may be composed of a few individual elements protectable under the substantial similarity standard should not imply that the substantial similarity standard applies to the work as a whole. Here, the individual elements protectable under the substantial similarity standard constitute such a small and isolated part of the entire work that substantial similarity should not engulf the much larger number of unprotectable items by providing the appropriate standard for the entire work.[137]

Acknowledging Judge Walker's good intentions with respect to the need to balance the competing considerations of the rights of the copyright holder against the risk of overprotection, one may make the following observations about the tests adopted by Judge Walker and the Ninth Circuit:

(a) The Appropriateness of the Test Where a License is Not Involved

Although the Apple v. Microsoft case has been one of the most widely watched look and feel cases from its inception, its ultimate precedential value for cases in which a license of some interface elements is not involved has always been uncertain. Because Judge Schwarzer determined early on in the case that the license Apple had granted Microsoft was a license of individual screen elements, Judge Walker – and derivatively the Ninth Circuit – focused his analysis of copyrightability and similarity on individual screen elements.[138] Where a license of

---

[136] Id. at 1447.

[137] Apple Computer, Inc. v. Microsoft Corp., No. C-88-20149-VRW (N.D. Cal. Apr. 14, 1993), at 4 (Order Establishing Hearing on Summary Judgment Motions, Pretrial Conference and Trial).

[138] The alleged similarities on the list supplied by Apple were, in part due to the court's prodding for details, mostly at a very detailed, individual element level, which further focused the court's analysis on elemental similarities.

individual elements is not involved, it is not clear that an analysis centered around individual elements – particularly as a root starting point – is appropriate.

There is reason to believe that other courts may adopt the general approach of performing analytic dissection of alleged similarities in the defendant's work as a first step in judging copyright infringement of a computer program user interface. As summarized further below in this article, a theme consistently appearing in the various forms of emerging infringement tests in both the "look" and the "feel" cases is that of "filtration" or "dissection" of a copyrighted work, at least as an initial step to determine what elements of the work in question are unprotectable. Given this trend, courts after Apple v. Microsoft may find Judge Walker's initial categorization step appealing even in cases where no license is involved that would force the court to focus on an element-by-element analysis. It remains to be seen, however, whether such courts will analyze the interfaces at issue in as fine a level of detail as the courts in this case did – that is, largely at the individual "widget" level.

Indeed, based on existing precedent, one can argue that analytic dissection at the widget level will be either inappropriate or unnecessary in many user interface infringement cases. One of the leading cases to date adopting a "filtration" approach is that of the Second Circuit in Computer Assocs. v. Altai Inc.,[139] which made filtration the explicit second step of a three step abstraction/filtration/comparison test for judging copyright infringement based on nonliteral structural similarities in the code of the program. As discussed in further detail in Part III below, the steps of the Second Circuit's test are:

> (1) abstraction of the work into levels of description with increasing specificity from the program's ultimate function down to its detailed line-by-line code;
>
> (2) filtration of structural components at each level of abstraction to determine whether their particular inclusion at that level was "idea" or was dictated by considerations of efficiency so as to be necessarily incidental to that idea, required by factors external to the program itself, or taken from the public domain, and hence non-protectable expression; and
>
> (3) comparison of the remaining "core of protectable expression" to determine whether the defendant copied any aspect of this protected expression.[140]

Although the Second Circuit stated that its test was intended for nonliteral structural infringement cases only, district courts in other Circuits have applied that test or extracts thereof to analyze "look," as well as "feel," elements in user interfaces.[141] Whether or not one believes

---

[139] 982 F.2d 693 (2d Cir. 1992).

[140] Id. at 706-11.

[141] See Gates Rubber Co. v. Bando American, Inc., 9 F.3d 823 (10th Cir. 1993); Autoskill, Inc. v. National Educational Support Sys., 793 F. Supp. 1557 (D.N.M. 1992), aff'd, 994 F.2d 1476, 1490-91 (10th Cir.), cert. denied, 114 S. Ct. 307 (1993). In affirming, the Tenth Circuit in Autoskill explicitly noted, "In this preliminary injunction appeal we need not

that the particular formulation of the Second Circuit is appropriate for a user interface case, the important point to note about that test is that it applies a levels of abstraction analysis to the works at issue before any filtration takes place.

Application of a levels of abstraction analysis first is both consistent with long-standing copyright infringement analysis, and appropriate for a user interface case. In particular, it seems that one should analyze the alleged similarities between two user interfaces at issue at the various levels of abstraction at which the works can be defined to first determine at which levels of abstraction similarities in fact (or allegedly) exist. This approach has two virtues. First, if there are no similarities below a certain level of abstraction, one need not determine which elements of the interface at such lower levels of abstraction are protectable. Thus, for example, if there are no similarities at the level of individual "widgets" of the interface, then one need not rule on which of those widgets are protectable.

Second, applying a levels of abstraction analysis first to determine at which levels similarities exist enables the court to determine which similarities are similarities of "idea" only – and therefore should be ignored by the finder of fact in judging infringement of the works overall – and which are potential similarities of "expression" that must be taken into account by the finder of fact when adjudicating infringement under an appropriate standard of similarity.

Neither Judge Walker's test nor the Ninth Circuit's formulation explicitly engages in a levels of abstraction analysis first. To the extent that Judge Walker's test considers higher levels of abstraction, it is primarily the later steps that look to whether copied individual elements "significantly alter" the works as a whole, or whether there is an "innovative melding" of unprotectable elements. As previously noted, however, the Ninth Circuit does not use these terms in its opinion, although the Ninth Circuit opinion certainly recognizes that expression at higher levels of abstraction may be protectable.[142]

The approach of not considering a levels of abstraction analysis first has at least two potential drawbacks, particularly for the ordinary case in which there has not been a license by the plaintiff of individual screen elements. First, it may require unnecessary analysis of the protectability of individual screen elements. Elements may not have been copied at the individual level, or the plaintiff may not be complaining about copying of a few individual elements. It therefore makes more sense to first determine the levels of abstraction at which the plaintiff is alleging copying occurred (which may be only "larger" elements of expression made up of nonsimilar, or unprotectable, individual elements), then determine as a matter of law whether those alleged similarities at that level of abstraction constitute "idea" or "expression," or are subject to a limiting doctrine, and then apply a similarity test.

---

decide which is precisely the correct method of analysis for a final copyright judgment ... because we are satisfied that the trial judge used a permissible method of analysis here".

[142] "This does not mean that at the end of the day, when the works are considered under the intrinsic test, they should not be compared as a whole. ... Nor does it mean that infringement cannot be based on original selection and arrangement of unprotected elements. However, the unprotectable elements have to be identified, or filtered, before the works can be considered as a whole." Apple v. Microsoft, 35 F.3d at 1446.

The second, related, drawback is that, by its focus first on detailed user interface elements – many of which are likely to be found unprotectable by themselves – the approach may tend to overemphasize the "trees" in derogation of a "forest" that may be the real matter at issue. "Larger" elements of expression are treated in the later stages of both Judge Walker's and the Ninth Circuit's test, after a myriad of individual screen elements – which may not form the real basis of the plaintiff's complaint – have been analyzed on a one-for-one basis. In the case where a license of individual screen elements is not involved, such element by element analysis may be both unnecessary and inappropriate. Application of a levels of abstraction analysis first should circumvent these problems.

(b) The Test for Similarity

Both Judge Walker and the Ninth Circuit attempt in their infringement tests to balance the competing considerations of the rights of the copyright holder against the risk of overprotection in part by adjusting the test for similarity to be applied. Both courts state that the choice between a substantial similarity test and a virtual identity test is to be based upon whether the work is capable of only a narrow range of expression. In application, however, as the flow charts in Figure 1 and Figure 2 demonstrate, the courts in this case determined which similarity test is to be applied based largely upon the nature of the individual elements that comprise the "expression" at issue. If those elements are largely "unprotectable" or not copied on an individual level, a virtually identical standard is applied. Otherwise, a broader standard is applied. The Ninth Circuit sets up a continuum between a case in which virtually all similarities relate to individually unprotectable elements (in which case only the original selection and arrangement of unprotectable elements is potentially protectable, to which a virtually identical standard of similarity is applied), and a case at the other end of the continuum in which most or many similarities relate to individually protectable elements (in which case the traditional substantial similarity standard is applied).

There is certainly precedent in the copyright law for adjusting the standard of infringement to be applied based on the range of potential expression inherent in the plaintiff's work.[143] However, the district court's and the Ninth Circuit's use of the individual elements comprising the allegedly copied portion of a work to determine the standard to be applied may again be unnecessary or inappropriate in many cases. If the plaintiff is not basing its claim on copying of individual elements, then it may be unnecessary and potentially inappropriate to have the standard of similarity turn on such elements.

Again, application of a levels of abstraction analysis seems the appropriate first step to judge which standard of similarity should apply. In particular, one should determine the levels of abstraction at which the plaintiff is alleging copying occurred, and determine as a matter of law whether those alleged similarities at each such level of abstraction constitute "idea" or "expression." For those that constitute "expression," one should then determine – for each level – whether that expression is capable of only a limited range of expression or is otherwise subject

[143] See, e.g.. Computer Assocs. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Manufacturers Technologies, Inc. v. CAMS, Inc., 706 F. Supp. 984 (D. Conn. 1989).

to one or more limiting doctrines. For those levels of abstraction at which only a limited range of expression is possible, or for which the expression is largely subject to one or more limiting doctrines, a standard of virtual identity should be applied to judge similarity of expression at each such level of abstraction. For those levels of abstraction not bounded by a limited range of possible expression and not subject to limiting doctrines, the traditional substantial similarity test should be applied to judge similarity of expression at each such level of abstraction.

The Ninth Circuit's "continuum" could potentially be applied consistent with this approach, although the Ninth Circuit's opinion does not spell out how the continuum should be applied to anything other than its two extremes. Further judicial development from the Ninth Circuit will therefore be necessary to understand the full meaning of the test the Ninth Circuit has set up.

E. THE BROWN BAG CASE

The Brown Bag case is one of the more important cases regarding look and feel to come out of the Ninth Circuit in recent years, for it transforms the legal test that had previously been applied to computer program look and feel claims.[144] The Ninth Circuit had laid the groundwork for this transformation in a case decided two years earlier, Shaw v. Lindheim.[145] Before Shaw, the Ninth Circuit had used an "extrinsic-intrinsic" test for judging copyright infringement, in which the court judged similarity of "ideas" in the "extrinsic" test and, if such similarity were found, the trier of fact decided whether the expression of the two works was substantially similar under a subjective analysis using an ordinary observer test.[146]

In Shaw, the Ninth Circuit transformed the "extrinsic-intrinsic" test into an "objective-subjective" analysis of expression. The court expressly limited its new test in Shaw to works of literature. Under the first step of the Shaw two-prong test, the court must judge whether the accused work is "objectively" similar in expression to the copyrighted work using analytic dissection, perhaps aided by expert testimony. If objective similarities are found, then Shaw held that subjective similarity of expression must be adjudicated by the finder of fact. "[T]he intrinsic [subjective] test cannot be the sole basis for a grant of summary judgment."[147]

The principal difference between the old "extrinsic-intrinsic" test and its reformulation in Shaw is that, under Shaw, both prongs of the test involve the analysis of expression, whereas under the old formulation the first step involved the analysis of only ideas, and the second step

---

[144] See generally Zimmerman, "Substantial Similarity of Computer Programs After Brown Bag," 9 The Computer Lawyer 6 (July 1992).

[145] 919 F.2d 1353 (9th Cir.), reh'g en banc denied, 1990 U.S. App. LEXIS 20420 (9th Cir. 1990).

[146] See also Sid & Marty Krofft Television v. McDonald's Corp., 562 F.2d 1157, 1164 (9th Cir. 1977).

[147] 919 F.2d at 1360.

involved the analysis of expression.[148] In Brown Bag, the Ninth Circuit applied the Shaw test to computer programs,[149] and held that where there was no showing of objective similarity of expression, summary judgment was appropriate for the defendant.

In Brown Bag, the plaintiff held the copyright on a computer program called "PC-Outline," which had been developed by an independent computer programmer named John L. Friend. PC-Outline was inspired by a preexisting outlining program owned by Symantec Corp. known as "ThinkTank." In 1987, Friend sold PC-Outline to Brown Bag, and Brown Bag granted back to Friend a license to use 129 pages of source code that generated certain user interface displays. Also in 1987, Friend developed and sold an outlining program called "Grandview" to Symantec. In 1988, Brown Bag brought suit against Symantec, alleging that Grandview infringed Brown Bag's copyright in PC-Outline.

In the district court proceedings, Brown Bag submitted a list of seventeen specific features in PC-Outline and Grandview alleged to be similar. The district court held that, because all alleged similarities were either licensed to Friend for use, were unprotectable under copyright, or were in fact not similar, summary judgment should be entered for the defendant.[150] Specifically, the district court ruled that the following alleged similarities could not form the basis for a finding of copyright infringement:

- The presence of four options in the main menu screen for accessing existing files, editing existing files, and printing. The court ruled that these concepts were "fundamental to a host of computer programs" and therefore not protectable.[151]

- Similarities in the nine functions listed in the menu bar of the program and the fact that virtually all of the functions of PC-Outline could be performed by Grandview, even though the functions were often labeled differently and accessed differently through the menus. The court held that these functions "constitute the idea of the outlining program," and that in any event the various fundamental functional features of the two programs were "quite different in their complementary features."[152]

- Use of pull down windows in the menu system. The court held that the plaintiff could "not claim copyright protection of an idea and expression that is, if not

---

[148] "Because the criteria incorporated into the extrinsic test encompass all objective manifestations of creativity, the two tests are more sensibly described as objective and subjective analyses of expression ...." Id. at 1357 (emphasis in original).

[149] "[C]omputer programs are subject to a Shawtype analytic dissection of various standard components, e.g., screens, menus, and keystrokes." Brown Bag Software v. Symantec Corp., 960 F.2d 1465, 1477 (9th Cir.), cert. denied, 113 S. Ct. 198 (1992) (emphasis in original).

[150] Telemarketing Resources v. Symantec Corp., 12 U.S.P.Q.2d (BNA) 1991 (N.D. Cal. 1989).

[151] Id. at 1995.

[152] Id.

standard, then commonplace in the computer software industry."[153] The court also ruled that in any event the pull down windows of the two programs looked different.

- Similarities in the color schemes of the two programs. The court rejected this, finding that fewer than 10 of the 44 default color selections in Grandview were the same as the default color selections in PC-Outline. The court also held that the similar blue backgrounds could not be a basis for finding infringement based on functionality of the color blue,[154] and a Copyright Office rule specifically excluding from copyright registration "typographic ornamentation, lettering or coloring."[155]

- Similarities of using the main editing screen to enter and edit data. The court found that the "need to have a screen from which a user can perform editing functions is essential to the very idea of a computer outline program."[156]

The court found it unnecessary to review the alleged similarities in the drawing of the menu bar, the use of highlighting bars and the use of a single line box around the menus because those features had been licensed to Friend to use.

On appeal, Brown Bag asserted that the district court improperly engaged in analytic dissection of the user interface of its copyrighted program, and that it neglected to evaluate the "overall look and feel" of the two programs for substantial similarity. The Ninth Circuit rejected these arguments, noting that analytic dissection of expression was proper under the reformulated "extrinsic" or "objective" test "for the purpose of defining the scope of plaintiff's copyright. ... To the extent a plaintiff's work is unprotected or unprotectable under copyright, the scope of the copyright must be limited."[157]

The Ninth Circuit also rejected Brown Bag's argument that the district court failed to look at the overall look and feel of the program in the intrinsic test. "Even assuming this is true, we find no reversible error because the record fails to include any evidence indicating that Brown Bag requested the district court to make this analysis. ... Therefore, there is no reason we should expect the district court to have made the analysis Brown Bag now requests."[158]

By dismissing Brown Bag's argument concerning the overall look and feel on the ground that it did, the Ninth Circuit avoided addressing the most vexing and important issue that the Data East, Paperback, and Apple decisions left open: once having identified unprotectable elements in a user interface through analytic dissection, how does one then treat those elements

---

[153] Id.

[154] Id.

[155] 37 C.F.R. § 202.1.

[156] 12 U.S.P.Q.2d at 1995.

[157] Brown Bag Software v. Symantec Corp., 960 F.2d 1465, 1476 (9th Cir.), cert. denied, 113 S. Ct. 198 (1992).

[158] Id..

in judging substantial similarity of the two works overall? Indeed, the Ninth Circuit in its opinion in Shaw explicitly acknowledged that it was not deciding this crucial question:

> The degree to which unprotected or unprotectable features must be eliminated from a comparison of two works is difficult to say. Although copyright protection is not afforded to certain elements of a work, such limitations "must not obscure the general proposition that copyright may inhere, under appropriate circumstances, in the selection and arrangement of unprotected components."[159]

By making reference to "selection and arrangement of unprotected components," the Ninth Circuit seems implicitly to acknowledge what the various Apple decisions, as discussed above, also seem to leave open as a possibility – that the combination or arrangement of individually unprotectable elements may form some larger "totality" of expression that is more than merely the sum of its parts. The Ninth Circuit, however, gives us no guidance in Brown Bag for judging when "overall look and feel" may be protected, and for separating cases in which summary judgment should be granted because all analytically dissected similarities are unprotectable and there is no larger "totality," from cases in which some larger "totality" must be judged by the trier of fact for substantial similarity, despite all individual similarities being unprotectable.

F. THE ATARI GAMES CASE

A decision from the D.C. Circuit, Atari Games Corp. v. Oman,[160] suggests that the threshold for finding copyrightable expression in a "totality" comprised of individually unprotectable elements is quite low. The Atari case reviewed a decision by the district court affirming the Register of Copyright's refusal to register Atari's video game "Breakout." Breakout was a very simple, early video game in which a player moved a rectangular "paddle" to hit a "ball" against a "brick wall" composed of eight rows of rectangles arranged in four monochromatic stripes (red, amber, green and yellow). When the square ball hit a rectangle, the rectangle vanished. When the ball broke through the wall of rectangles to the empty space beyond, it ricocheted at increased speed until reemerging. The ball's movement did not follow the laws of physics – instead, the angle of the ball's rebound depended solely on where it impacted the paddle.

The Register refused to register the game on the ground that it was too trivial for protection, that it was composed entirely of unprotectable elements such as squares and other geometric shapes, and that there was no distinctive arrangement or unique graphic design in the game. The district court affirmed the Register's dismissal.[161]

---

[159] Id. at 1476 n.4 (quoting II M. Nimmer & D. Nimmer, Nimmer on Copyright § 13.03[F][5], at 13-78.44 n.342, and citing Feist Publications, Inc. v. Rural Telephone Serv. Co., 111 S. Ct. 1282, 1289, 1294 (1991)).

[160] 979 F.2d 242 (D.C. Cir. 1992).

[161] Atari Games Corp. v. Oman, 8 U.S.P.Q.2d 1426 (D.D.C. 1988).

The Second Circuit reversed, holding that the Register had focused too much on individual screens of the game, and had given insufficient treatment to the arrangement of screens and "the flow of the game as a whole."[162]

> We can accept the Register's assertion that the individual graphic elements of each screen are not copyrightable. Even so, BREAKOUT would be copyrightable if the requisite level of creativity is met by either the individual screens or the relationship of each screen to the others and/or the accompanying sound effects.[163]

Citing Feist,[164] the court also held that the Register required too high a standard of originality in stating that Breakout contained no elements of graphic design that were "unique" or "distinctive." The court noted that Feist excluded from protection only those arrangements that were "mechanical," "garden-variety," "typical" or "obvious," or followed a "convention" that is "purely functional."[165] The court found that Breakout's arrangement of graphic elements in individual screens, and its sequence of screens, exceeded the threshold specified in Feist. In particular, the court found that the choice of having a ball that does not follow the laws of physics was not obvious, and the coordination of a square ball and a rectangular shrinking paddle was not "conventional."[166]

The court also noted that the same could be said of "the choice of colors (not the solid red, brown, or white of most brick walls), the placement and design of the scores, the changes in speed, the use of sounds, and the synchronized graphics and sounds which accompany the ball's bounces behind the wall."[167] Accordingly, the court reversed the district court's grant of summary judgment to the Register and remanded Atari's application to the Register for renewed consideration consistent with the court's opinion.

The Atari case lends support to the notion that, even though a work may be comprised of very simple, commonplace, and individually unprotectable elements, there may nevertheless be protectable expression in the "totality" of the work's arrangement of elements within a screen and its sequence of screens. The court implied in its opinion, however, that where the elements forming the "total" expression are largely unprotectable, the copyright may afford protection against only virtually identical copying.[168]

---

[162] 979 F.2d at 245.

[163] Id. at 244.

[164] Feist Publications v. Rural Tel. Serv. Co., 111 S. Ct. 1282 (1991).

[165] 979 F.2d at 246.

[166] Id. at 247.

[167] Id.

[168] "A determination of copyrightability, we note, does not mean the holder will prevail against all copiers or producers of similar works. The scope (strength or 'thinness') of the protection is a distinct inquiry." Id. at 244 n.4 (quoting the following passage from Frybarger v. International Business Machines Corp., 812 F.2d 525, 529-30 (9th Cir. 1987): "the mere

G. THE CAPCOM v. DATA EAST CASE

A very interesting recent video game case, Capcom U.S.A., Inc. v. Data East Corp.,[169] raised significant issues in both the "look" and the "feel" areas. On the "look" side, this case raised all the questions left open by the Ninth Circuit's decision in Apple v. Microsoft concerning how one handles unprotectable elements in presenting a copyright infringement case to a jury. Because the case settled shortly before trial, the court never had an opportunity to rule on a motion brought by Data East to clarify an earlier ruling of the court that Data East believed would exclude at least many of the unprotectable elements from being presented to the jury, nor did the court have an opportunity to decide the jury instructions that would be presented to the jury. On the "feel" side, Data East obtained a significant ruling that the joystick control sequences for invoking various moves of the game characters were uncopyrightable.

1. Factual Background

Capcom, Ltd. is the developer of the "Street Fighter" video game series. Capcom U.S.A., Inc. is the exclusive distributor of these games in the United States. Although the original "Street Fighter" game was not particularly successful, the "Street Fighter II" series of games ("SFII"), first introduced in 1991, became one of the most successful video game series of all times, selling over 50,000 arcade versions and over two million home versions in the United States. In 1993, the defendant Data East Corp. released a competing video game called "Fighter's History," which was in the same genre of games as SFII.

Both SFII and Fighter's History were one-on-one, martial arts-based fighting games which pitted two characters against one another.[170] SFII had twelve fighting characters, eight of which the player could select and control for game play and four of which were controlled by the computer. Fighter's History had nine characters, seven of which the player could select and control and two of which were controlled by the computer. Each game could be played in one-player mode, with the computer controlling the opponent, or in two-player mode. Each character in the games had a particular personality and martial arts fighting style containing a variety of moves. The moves included "common moves" (simple punches and kicks based largely on standard martial arts techniques), "special moves" (fantasy-based moves, not derived from real fighting techniques), and "combination attacks" (a series of special moves executed in rapid sequence that, if properly executed, cannot be blocked once the first move strikes the opponent).

In both games, the characters' moves and defenses were controlled by an eight position joystick, and three kick and three punch buttons of varying intensities. The characters' special moves were invoked by certain distinct control sequences of joystick movements and kick and punch button presses. In both games, the fighter who won two-out-of-three rounds won the

---

indispensable expression of these ideas ... may be protected only against virtually identical copying" (emphasis deleted)).

[169] No. C 93-3259 WHO (N.D. Cal. Mar. 16, 1994).

[170] The first video game in this genre of games was Data East's "Karate Champ" game, which was the subject of the Data East v. Epyx case discussed above.

fight. Each round was won by depleting the opponent's "vitality bar" or having more vitality remaining when the round's time expired. A player could deplete an opponent's vitality in varying amounts by striking the opponent with common moves, special moves and/or combination attacks. Both games displayed a round timer at the top of the screen that counted down from 99 to 0.

Capcom filed suit against Data East, alleging that Fighter's History infringed Capcom's copyrights in SFII and also infringed SFII's trade dress and trademarks.[171] With respect to its copyright claim, Capcom identified a number of alleged similarities between the games in four primary categories: (1) similarities in characters; (2) similarities in special moves and combination moves; (3) similarities in control sequences; and (4) miscellaneous similarities in the general presentation and flow of the games.

2. The Motion for a Preliminary Injunction

Capcom moved for a preliminary injunction against Data East. Data East opposed the motion on the ground that all of the similarities between the games identified by Capcom were either unprotectable by copyright law or were in fact not similar as a matter of law. Data East submitted a large volume of evidence drawn from elements of popular culture such as comic books, movies, posters, other video games, books, and the like, to demonstrate that a great many of the alleged similarities between the characters and their moves were in fact not original to Capcom and were preexisting stereotypes constituting scenes a faire.

After an extensive hearing with expert testimony, the court denied Capcom's motion for a preliminary injunction, finding that Capcom had showed neither a likelihood of success on the merits nor that the balance of hardships tipped in its favor.[172] The court noted that under the Brown Bag case, a two-step analysis must be performed. First, the two works must be compared under the "extrinsic test" using an objective analysis of similarity, perhaps aided by expert testimony. Under the extrinsic test, the court engages in analytic dissection of the alleged similarities to determine whether such similarities stem from unprotectable or protectable expression. Second, one then applies an "intrinsic test" of subjective similarity to ascertain whether there exists similarity of protectable expression.[173]

---

[171] Capcom alleged that its entire game constituted a protectable trade dress that was infringed by Fighter's History. Data East moved for summary judgment on the trade dress claims on the legal ground that Capcom's *game itself*, as distinct from the arcade box, packaging, and "attract" screens in which the game was presented to prospective players, was not protectable as a matter of law under the doctrine of functionality and other doctrines. The case settled before the court had an opportunity to rule on Data East's motion.

[172] Under Ninth Circuit law, to obtain a preliminary injunction, a party must show either (1) a likelihood of success on the merits and the possibility of irreparable injury, or (2) that serious questions going to the merits exist and the balance of hardships tips sharply in its favor. Johnson Controls, Inc. v. Phoenix Control Sys., 886 F.2d 1173, 1174 (9th Cir. 1989).

[173] Capcom U.S.A., Inc. v. Data East Corp., No. C 93-3259 WHO, slip op. at 10-11 (N.D. Cal. Mar. 16, 1994) (citing Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp. 616, 623 (N.D. Cal. 1993)).

Citing one of Judge Walker's opinions in Apple v. Microsoft, the court stated that in the extrinsic test "[a] court must filter out those elements of the copyrighted work that are deemed unprotectable, and reserve only protectable expression for comparison under the subjective test."[174] The court further noted that "the analytic dissection serves two functions: (1) it first determines the parameters of the copyrightable expression; and (2) the dissected, protectable elements can then be compared individually with the corresponding elements of the challenged work to detect similarities."[175] The court noted that the Ninth Circuit had identified several categories of unprotectable elements that must be filtered out of the analysis using analytic dissection under the extrinsic test: ideas, merged expression, scenes a faire, and functional aspects of a work that constitute methods of operation, procedures and processes.[176]

(a) Similarities in Control Sequences

The court then turned to an analytic dissection of the four categories of alleged similarities identified by Capcom. With respect to alleged similarities in control sequences (joystick and button combinations that a player must execute to invoke particular moves), Capcom argued that such control sequences were copyrightable as "nonliteral components" of the user interface of the video game computer program. The court rejected this argument, ruling that the control sequences were not copyrightable on three grounds:

- Merger: "[T]he expression of an idea and the underlying idea frequently merge in the area of control sequences because the player simply presses the button corresponding to the move he wishes to have produced on the screen."[177]

- Functional and Practical Constraints: "[T]he use of the joystick is functionally constrained because there are only eight possible positions the joystick can occupy. Moreover, the technique that a developer uses in creating joystick sequences is further constrained by the fact that the movements must be connected – the joystick can move only from one position to a neighboring position. ... On the practical level, the universe of possible joystick combinations is further restricted by the need to have the control sequences emulate the natural movements of the body. To make the game realistic and easy to learn, a developer must have its control sequences follow the natural flow of the fighter's body."[178]

---

[174] Slip op. at 11.

[175] Id. at 12 (emphasis added). The court's reference to a comparison "individually" with the corresponding elements of the challenged work suggests that one looks only at the protectable elements of the work in making the comparison, and not the works as a whole, including the unprotectable elements. The issue of what elements one looks to in making the comparison of the intrinsic test is discussed further below.

[176] Id. at 12-13.

[177] Id. at 14-15.

[178] Id. at 15. Capcom pointed to a number of control sequences in Fighter's History that were identical to SFII that allegedly did not emulate the body's natural movement, but instead were

- Useful Article Doctrine: "[U]nder the law of the Ninth Circuit, an article that has 'any intrinsic utilitarian function' is ineligible for copyright protection[] 'except to the extent that its artistic features can be identified separately and are capable of existing independently as a work of art.' ... Here, there are no identifiable artistic features in the control sequences that are separable from their functional purpose."[179]

(b) Similarities in Miscellaneous Game Features

Capcom alleged a number of similarities in miscellaneous features in the general presentation and flow of the games, including the "attract mode", the "VS." screens at the beginning of the game in which a player selected his character from a row of faces and the faces of the two selected characters then were shown facing one another prior to commencement of the round, designating winners by showing the opponent's face bloodied at the end of a game, and tracking a player's vitality during a fight using a yellow and red horizontal vitality bar at the top of the screen. The court ruled that neither these miscellaneous features, nor their particular compilation in SFII, were protectable because they constituted scenes a faire. "Indeed, several of the non-infringing games that Capcom submitted to the Court for purposes of comparison include these same features."[180]

(c) Similarities in Characters and Special Moves

With respect to alleged similarities in characters and special moves, the court noted that, to the extent the characters embodied preexisting stereotypes, they were unprotectable, and only the "concrete details of the visual presentation constitute the copyrightable expression."[181] The court found that a number of the allegedly similar special moves were unprotectable because they were nothing more than common moves from kickboxing, wrestling and the like that embodied no "expressive detail above the basic idea that they represent."[182] The court eliminated two of the eight SFII characters at issue because there were no Fighter's History characters similar to them.[183] After a detailed analysis of all remaining alleged similarities in characters and special moves, the court found that "of the eight pairs of characters and twenty-seven special moves at issue, three characters and five special moves in Fighter's History are similar to protectable characters and special moves in Street Fighter II. These figures must be cast against the fact that Street Fighter II has a total universe of twelve characters and six

---

wholly arbitrary. The court noted that, while it was "disturbed by these 'coincidences' in some of the arbitrary control sequences, it concludes that because the control sequences do not constitute protectable expression, these isolated similarities are not actionable." Id. at 17.

[179] Id. at 16 (quoting Fabrica, Inc. v. El Dorado Corp., 697 F.2d 890, 893 (9th Cir. 1983)).

[180] Id. at 17.

[181] Id. at 20 (quoting Atari v. North Am. Philips, 672 F.2d 607, 617 (7th Cir.), cert. denied, 459 U.S. 880 (1982)).

[182] Slip op. at 24.

[183] Id.

hundred and fifty moves. Capcom concedes, as it must, that the vast majority of the moves are unprotectable because they are commonplace kicks and punches."[184]

(d) Application of the Intrinsic Test

The court, acting as the fact finder on the motion for a preliminary injunction, then turned to an application of the intrinsic test to the protectable elements. "[O]nce all the unprotected similarities and those elements that are not similar as a matter of law have been filtered out through analytic dissection, the remaining protectable expression in the plaintiff's video is compared to corresponding expression in the defendant's video in a subjective analysis of similarity."[185] The court's reference to "corresponding" expression is interesting, as it suggests a comparison at an individual element-by-element level first, much the same as in the case of both Judge Walker's test and the Ninth Circuit's test in the Apple v. Microsoft case illustrated in Figures 1 and 2 above.

Again citing one of Judge Walker's opinions in Apple v. Microsoft, the court ruled that "[w]here the alleged similarities in a plaintiff's work consist[] primarily of elements that are unprotectable, or are capable of only a narrow range of expression, then a court must apply a 'virtual identity standard' when comparing the plaintiff's work with the challenged one."[186] The court noted that it was "indisputable that Street Fighter II is largely comprised of unprotectable elements. The vast majority of the moves – over 650 of them – are unprotectable, commonplace punches and kicks. In addition, the Court finds that even a majority of the moves that are allegedly special and fanciful are ultimately unprotectable either because they are unoriginal scenes-a-faire or have not actually been copied by Data East. As a result, the virtual identity standard is the appropriate standard for the Court to apply in assessing the subjective similarity between the two games."[187]

The court then noted that the "subjective determination involved in the intrinsic test employs a reasonable person standard and examines the works for similarity in 'total concept and feel.'"[188] The court's reference to "total concept and feel" suggests some kind of comparison of the works as a whole. Thus, although the court did not elaborate, it appears that the court felt that the adjudication of similarity under the intrinsic step must be made at both an individual, element-by-element, level and a works-as-a-whole level. The court did not, however, detail precisely how the unprotectable elements were to be handled in comparing the works as a whole.

---

[184] Id. at 25.

[185] Id. at 25-26 (emphasis added).

[186] Id. at 26. The court also cited for this proposition Harper House, Inc. v. Thomas Nelson, Inc., 889 F.2d 197, 205 (9th Cir. 1989).

[187] Slip op. at 26.

[188] Id. at 26-27.

Applying the virtual identity standard, the court found that "Data East has not copied the core, protectable expression in Street Fighter II."[189] With respect to individual similarities, the court found that, to the extent that similarities did in fact exist, none of the characters or special moves that had allegedly been copied had been "bodily appropriated." The court further ruled: "Even if the Court were to apply the more lenient substantial similarity standard, its conclusion would be the same. The physical appearances and fight moves of the few characters at issue are expressed in sufficiently different manners in the two games to preclude a finding of substantial similarity."[190]

With respect to a comparison of the works as a whole, the court ruled:

> In the end, Data East has not captured in Fighter's History the "total concept and feel" of the protectable expression in Street Fighter II. Rather, the similarities that result between the two games stem from Data East's emulation of the unprotectable, commonplace features of Street Fighter II, such as its stereotypical fight characters and its reliance on unoriginal fighting techniques derived from the martial arts.[191]

The court's reference to "protectable expression" in judging the "total concept and feel" suggests that the unprotectable elements must be filtered out or ignored in the analysis of the works as a whole, although the court does not elaborate on this point. The court therefore concluded that Capcom had failed to raise serious questions concerning the merits of its copyright claim, and, accordingly, denied the motion for a preliminary injunction.

3. The Motion for Summary Judgment

Following the court's denial of a preliminary injunction, Data East moved for summary judgment with respect to all of Capcom's copyright claims. Data East argued that the court's legal rulings under the extrinsic test in its opinion denying a preliminary injunction, the undisputed facts concerning the audiovisual elements produced by the games, and the undisputed facts concerning other elements of popular culture that existed before SFII, were sufficient to dispose of Capcom's copyright claims as a matter of law.

Capcom opposed the motion on three basic grounds. First, Capcom argued that because the court found under the extrinsic test that several of Fighter's History characters and moves were similar to protectable expression in SFII, then, at a minimum, these similarities created a triable issue for the jury. Second, Capcom maintained that because the court identified similarities between Fighter's History and several protectable elements in SFII under the extrinsic test, then the entire game, including both protectable and unprotectable expression, must be submitted to the jury for analysis of similarity in "total concept and feel" under the intrinsic test. Third, Capcom argued that the court's rulings under the extrinsic test were all

---

[189] Id. at 27.

[190] Id. at 28.

[191] Id. at 29 (emphasis added).

premised on disputed factual issues, the resolution of which was not appropriate for summary judgment.

The court granted Data East's motion in part, and denied it in part. The court refused to grant Data East summary judgment with respect to all of Capcom's copyright claims on the ground that there were factual disputes to be resolved by a jury on the issue of similarity. Specifically, the court ruled that under Ninth Circuit law, because the court had found in its preliminary injunction ruling that Fighter's History contained three characters and five special moves that were similar to protectable expression in SFII, Capcom was entitled to a jury trial at least with respect to those eight elements of similarity.[192]

In addition, the court ruled that its determination at the preliminary injunction stage that other allegedly similar characters and special moves were more dissimilar than similar was based on conflicting testimony concerning similarity, and therefore raised disputed factual issues that must be resolved by the jury. The court stated that its determination at the preliminary injunction stage that certain characters and moves were more dissimilar than similar "does not mean that no reasonable juror could disagree with the Court's outcome on the question of substantial similarity and, consequently, does not mean that summary judgment is appropriate with respect to these elements. The jury should now be given the opportunity to make the determination of similarity anew."[193]

The court rejected, however, Capcom's argument that once a court finds that there is objective similarity of protectable expression under the extrinsic test, then both the protectable and the unprotectable elements of the two works must be compared as a whole by the jury under the intrinsic test's "total concept and feel" standard:

> A review of the Ninth Circuit case law on this point, however, reveals ample support for the Court's decision to consider only protectable expression in its intrinsic analysis. [Discussion of citations omitted]
>
> Finally, and as a matter of logic, if the Court were to accept Capcom's argument that both the protectable and unprotectable elements of a work can be compared for similarity in "total concept and feel" under the intrinsic test, then the dissection required under the extrinsic test would be rendered meaningless. There would be no point in performing analytic dissection to separate the protectable elements from the unprotectable ones during the extrinsic test as the Ninth Circuit clearly requires, if, in the end, courts were free to compare the two works in their entirety under the intrinsic test.[194]

---

[192] Capcom U.S.A., Inc. v. Data East Corp., No. C 93-3259 WHO (N.D. Cal. Aug. 18, 1994), at 7-10.

[193] Id. at 18.

[194] Id. at 10, 12.

In addition, the court granted Data East partial summary judgment with respect to several specific alleged similarities:

(a) Similarities in Miscellaneous Game Features

The court reaffirmed its previous ruling that the similarities in miscellaneous game features (such as the attract mode, the "VS." screens and the methods used for selecting players, tracking vitality and designating winners) were commonplace in the videogame industry and, consequently, unprotectable scenes a faire. The court noted that Capcom had presented insufficient evidence to rebut this determination, and therefore drew the following conclusion: "As such, there is no triable issue of fact with respect to the unprotectable nature of Street Fighter II's miscellaneous game features and Data East is entitled to summary judgment with respect to this category of alleged similarities.[195]

(b) Similarities in Control Sequences

With respect to control sequences, the court noted that its earlier determination that the control sequences were not protectable "was almost purely legal in nature and, although Capcom may disagree with the outcome of the Court's legal analysis, this does not create a triable issue of fact for the jury. Data East is entitled to summary judgment with respect to the alleged similarities in control sequences between the two games."[196]

4. The Motion for Clarification

In response to the court's ruling rejecting Capcom's argument that, once a court finds objective similarity of protectable expression under the extrinsic test, then both the protectable and the unprotectable elements of the two works must be compared as a whole by the jury, Data East brought a motion for clarification of the court's summary judgment ruling, asking the court to clarify that its grant of partial summary judgment to Data East with respect to several alleged similarities meant that Capcom could not rely on such similarities as a basis for infringement, and such similarities should therefore not be presented to the jury.

Data East's motion for clarification was brought for two primary reasons. First, it became apparent during continuing discovery after the court's ruling on Data East's motion for

---

[195] Id. at 14.

[196] Id. at 14-15. With respect to the issue of alleged similarities in combination attacks, the court set forth only the following, curious footnote: "In addition, and as a threshold matter, the Court determined that, unlike Street Fighter II, Fighter's History did not have preprogrammed 'combination attacks.' This determination was based on sharply conflicting expert testimony and declarations. The import of this determination was that it prompted the Court to separate combination attacks and examine their individual components to assess protectability and similarity, rather than viewing the combination as a whole. This more segmented analysis clearly impacted some of the Court's conclusions that individual elements of a Street Fighter II combination attack were not protectable or, if protectable, were not actually copied in Fighter's History." Id. at 17 n.2.

summary judgment that Capcom still intended to obtain discovery and present evidence to the jury with respect to allegedly copied elements of SFII that the court had ruled were not protectable. Data East wanted to narrow both the scope of remaining discovery and the issues and evidence that would be presented to the jury at trial. Second, Data East felt that clarification on the point would assist the parties and the court in preparing and approving jury instructions. The jury instructions would obviously differ greatly, depending upon whether or not some or all of the unprotectable elements of SFII were to be presented to the jury.

Capcom opposed Data East's motion, arguing (among other things) that, notwithstanding the court's rulings in the summary judgment motion, Capcom was entitled for six reasons to present to the jury elements in the two games that the court had determined were not copyrightable. Specifically, Capcom made clear its intent to present the entire games to the jury, including evidence of alleged similarities in both protectable and nonprotectable elements. Capcom's six reasons for arguing that it should be allowed to do so were as follows:

1. Overall Compilation. Capcom argued that it should be entitled to offer evidence regarding every element of the game to prove that SFII is a unique combination of separate elements and that Fighter's History copied that unique combination.[197] The jury would then be instructed regarding how they were to determine liability for infringement – e.g., by being told that only certain combinations of elements constituted copyrightable expression, and not the elements themselves (standing alone).[198]

2. Proof of Copying. Capcom argued that evidence regarding the unprotected elements should be admitted to prove intentional copying of protected elements of SFII.[199]

3. Overall Look and Feel. Capcom argued that in a case such as this one involving artistic, non-factual works of entertainment, "the jury is entitled to consider whether the 'look and feel' of the works as a whole are substantially similar and is not limited to an atomized and isolated look at only the protectable individual similarities."[200]

4. Context. Capcom argued that "evidence concerning the unprotected elements is part of the context for the subjective perception determination that the jury must make under the 'intrinsic' test for copyright infringement."[201]

---

[197] See Plaintiff Capcom's Opposition to Data East's Motion for Clarification of the Court's Order Dated August 18, 1994 (Oct. 27, 1994), at 17.

[198] "The jury presumably will receive instructions on what constitutes expression, originality, and other aspects of copyright law, and apply those instructions to all of the evidence in the case. The Court may or may not decide that various more specific instructions about what aspects of the games are or are not protected by copyright may be appropriate in the context of all the jury instructions and the evidence, as contemplated by Harper House." Id. at 21.

[199] Id. at 18.

[200] Id. at 18-19.

[201] Id. at 19.

5. Two-Supplier Market Theory of Damages. With respect to the unprotectable control sequences, Capcom argued that "evidence regarding copying of and similarity between control sequences is admissible to prove Capcom's lost profits under a 'two-supplier' market theory of damages."[202] Capcom apparently intended to rely on alleged copying of SFII's control sequences to prove that Data East "targeted" SFII players, who were familiar with such control sequences and could therefore play Fighter's History with relatively little retraining, to establish "a presumption that Fighter's History sales displaced Street Fighter sales."[203]

6. Likelihood of Confusion. Finally, Capcom argued that "elements of the games that of themselves are not protectable by copyright are admissible to prove the likelihood of confusion on Capcom's trade dress and unfair competition claims."[204]

Before the court could rule on Data East's motion for clarification, the case settled. Accordingly, the court never had an occasion to determine whether unprotectable elements could be presented to the jury, and if so, what the scope of such presentation would be and how it would be handled. Nor did the court have an opportunity to decide the jury instructions that would be presented to the jury, and specifically how the jury would be instructed to treat unprotectable elements in making its similarity comparison, if some or all of the unprotectable elements were allowed to be presented to the jury.

To date, the Capcom v. Data East and Apple v. Microsoft cases have been the two best look and feel cases to present a focused opportunity to answer these questions concerning how one handles elements in the adjudication of similarity that have been deemed unprotectable in an analytic dissection (or "filtration") step. Unfortunately, both cases ultimately failed to resolve the issue because of preempting dispositions that prevented them from ever reaching a jury trial.

H. THE INTERACTIVE NETWORK CASE

In the case of Interactive Network, Inc. v. NTN Communications, Inc.,[205] NTN Communications sought to prevent its competitor, Interactive Network, from offering a competing interactive football game having a similar scoring system, data feed structure, and user interface elements. NTN developed an interactive game to be played in conjunction with

---

[202] Id.

[203] Id. at 20.

[204] Capcom also argued that, on a purely practical level, it made no sense to try to exclude evidence of uncopyrightable similarities. "For example, one can't blot out the vitality bars, remaining time, rounds won, announcers' voices, or the control sequences if the games are demonstrated or films of game play are shown to the jury." Id. at 20. Capcom's argument on this point seems, however, to have presumed the answer to the legal issue at stake, for the very issue on which clarification was sought was whether the jury should be entitled to see a demonstration of the entire game or a film of game play at all.

[205] 875 F. Supp. 1398 (N.D. Cal. 1995).

televised football games called "QB1." Interactive sold QB1 for a while, and then developed its own competing game called "IN the Huddle." Interactive filed suit against NTN seeking a declaration that IN the Huddle did not infringe QB1.

NTN asserted that Interactive copied protectable expression of QB1 in the form of its three-level play prediction structure; the scoring system (including the "Game Breaker" feature, the "consecutive bonus" feature, and the 100/-10 point scoring system); the layout and arrangement of the graphical user interface; various other nonfunctional features such as the broadcast of an expert's score and statistical information and the use of a particular opening animation sequence; and NTN's real-time data feed, which was used to transfer coded information regarding play outcomes.

Applying the "extrinsic" test of the Ninth Circuit, the court defined the idea underlying the interactive games at issue, and then identified unprotectable elements that should be filtered out of the analysis of similarity in the "intrinsic" test. The court defined the idea underlying both games as "that of a live, play-along, interactive game to test one's ability to make multi-staged predictions regarding plays in a football game where one is rewarded with points corresponding to the difficulty and accuracy of the prediction made."[206] This definition of the idea was at a fairly high level, which left many of the details of NTN's games potentially protectable.

The court next applied the various limiting doctrines of copyright law to identify which of the alleged similarities were not protectable. The court concluded that there were only a limited number of ways of organizing play predictions for football. Accordingly, NTN's three-level prediction scheme was not protectable under the merger doctrine. The court also held that the awarding of points for successful actions and subtracting of points for mistaken actions, and the awarding of higher points for successful actions that are statistically less likely to occur, were unprotectable aspects of NTN's scoring system under the scenes a faire doctrine. However, the court ruled that the particular point values assigned to various correct and incorrect predictions were somewhat arbitrary (such as QB1's 100/-10 point scheme for a correctly or incorrectly predicted pass),[207] and that the "consecutive bonus" and "clutch pick" bonus did not necessarily derive from the idea of an interactive football game. Accordingly, those aspects of the scoring system, which had been identically copied by Interactive, were protectable expression, and the court denied summary judgment on behalf of Interactive as to those elements.[208]

NTN also argued that IN the Huddle needlessly copied the arbitrary ordering and structure of the data fields of NTN's data feed format. The court ruled that QB1's data feed

---

[206] Id. at 1404.

[207] The court noted that probabilities of plays derived from actual NFL statistical data could justify similarities in the relative scores awarded for various correct predictions. However, the evidence did not establish that those elements of QB1 copied in IN the Huddle were only of such nature. Id. at 1405.

[208] Elements of the games inherent in the sport of football, such as predictions of running and passing, to the left, middle, or right, and deep, short, or back, were also ruled not protectable. Id.

structure constituted protectable subject matter, relying on the Fifth Circuit's decision in Engineering Dynamics, Inc. v. Structural Software, Inc.[209] The court ruled, however, that as a matter of law, NTN had not submitted any evidence establishing substantial similarity between the data feed structures of the two games.[210] The court's ruling that the data feed format of NTN's program is protectable subject matter is an interesting one. NTN argued that copying of such data feed format was needless, apparently because there was no need for Interactive's game to be compatible with QB1. It is unclear whether it would have affected the court's ruling that the data feed format was protectable if Interactive had needed to copy such format for some technical reason of compatibility.[211]

The Interactive Network case is a further illustration of the risk competitors run in identically copying various aspects of a copyrighted program, at least where the defendant cannot demonstrate that such copying was dictated by technical compatibility reasons.

I. THE PRODUCTIVITY SOFTWARE CASE

The 1995 case of Productivity Software Int'l v. Healthcare Technologies, Inc.[212] raised the question of copyright protection for non-literal elements of add-in programs. The plaintiff Productivity Software marketed a computer program called "Productivity Plus" or "PRD+", which was an add-in program to a word processing program that allowed the typist to automatically expand short forms typed on the keyboard (such as "ny") into the proper words or phrases (such as "New York"). The defendant marketed a text replacement program named SHORTCUT which Productivity Software alleged infringed non-literal elements of its program. Productivity Software made no claim for infringement of its source code.

The district court adopted the abstraction/filtration/comparison analysis of the Second Circuit's Altai decision. Applying the abstractions step, the court defined the plaintiff's program as one designed to allow the user to search for and view short forms and their corresponding long forms using a menu screen, and to edit the short forms and the long forms using an edit screen. The court noted, however, that because the plaintiff's program was close to that of a simple form, it was entitled to only a narrow range of copyright protection.[213]

---

[209] 26 F.3d 1335 (5th Cir. 1994).

[210] 875 F. Supp. 1398, 1405 (N.D. Cal. 1995). The court also ruled that, as a matter of law, Interactive had not infringed NTN's copyright as a result of alleged similarities in QB1's graphical user interface, screen displays, and opening animation, because Interactive, not NTN, was the author of those features. Id. at 1405-06.

[211] The court also rejected a claim by NTN that the allegedly copied elements of its game constituted a protectable trade dress, because the court found such elements to be functional, both individually and in the arrangement and composition of them, as they constituted the "actual benefit that the consumer wishes to purchase, as distinguished from an assurance that a particular entity made, sponsored, or endorsed the a product." Id. at 1409.

[212] 37 U.S.P.Q.2d 1036 (S.D.N.Y. 1995).

[213] Id. at 1039.

Applying the filtration step of the test, the court looked at the non-literal elements of the plaintiff's PRD+ program at each level of abstraction. First, the court ruled that the basic abstraction of increasing a typist's efficiency by automatically replacing abbreviated words or phrases with their long forms was an unprotectable idea.[214] The plaintiff contended, however, that its main menu screen contained non-literal elements protectable by copyright – specifically, its placement of the short form list to the left of the long form list (which the plaintiff contended was counter-intuitive because a typist would normally know the long form and would therefore be searching for it in the long form list, which should be on the left).[215]

The court rejected this argument, ruling that placement of the abbreviated forms to the left was dictated by efficiency concerns, as this is the way abbreviated forms of words are placed in dictionary listings. The court further held that such "method of organizing lists of abbreviations is so closely linked with the underlying idea for the program as to be barred by the doctrine of merger" and that since considerations of utility dictate that the two lists be placed side by side, the choice of which list goes to the left may not be copyrighted if there is only one other option.[216] The court also found no original authorship in an unadorned two-column alphabetical listing.[217]

For similar reasons, the court rejected the plaintiff's argument that the defendant's moving of its menu bar from the bottom of the menu screen, where it appeared in the PRD+ program, to the top of the menu screen in its own program nevertheless constituted an infringing similarity. The court noted that there are only two locations where a menu bar may logically be placed on a computer screen (the top and the bottom), and such limited number of available alternatives prevented the plaintiff from claiming copyright protection for its placement of the menu bar.[218]

The court also rejected as a basis for infringement a number of other alleged similarities of non-literal elements of the menu screen, all of which the court found dictated by efficiency or common features:

- The dedication of a single line for each entry in the lists of short forms and long forms – dictated by efficiency because longer entries would complicate the display and interfere with searches through the lists.

- Use of a cursor to scroll through the entries in the program.

---

[214] Id. at 1040.

[215] Id. The plaintiff did not claim infringement of the items contained in the lists in the menu screen, however.

[216] Id. at 1040.

[217] Id.

[218] Id.

- Outlining of the short form and long form lists with a single line border, which the court found to be a common design element.[219]

Similarly, the court rejected as unprotectable a number of similarities in the programs' edit screens:

- Use of a separate edit screen of itself, which the court held not protectable since numerous computer programs utilize one screen configuration to display data and another screen configuration to input and edit data.

- Placement of short forms in a long narrow box at the top of the screen and the long forms in a larger box below, which the court found unprotectable because dictated by requirements of functionality and because a screen with one box for the short form and one box for the long form presented only two possible configurations with a horizontal arrangement, and only two additional configurations in a side-by-side arrangement.[220]

The court therefore concluded that each of the non-literal similarities identified by the plaintiff, taken by themselves, were not protectable because they were an idea, not original, or dictated by efficiency or external factors.[221] The court noted, however, that the plaintiff could still potentially claim protection in the combination of the unprotectable elements, although such combination would be entitled to only narrow protection. But the court concluded that no reasonable fact-finder could find the two programs' display screens to be substantially similar overall. The two main menu screens utilized different menu bar commands ("Add Edit Delete Name Gate Lift Record File Select Toggle Options" versus "File Edit Search Tools Help Search String"). And the defendant's program had a number of features not present in the plaintiff's program: pull down menus; context-sensitive help; execution of commands from the main menu without exiting the foreground application; auto search mode; and the ability to turn off the expansion function while in the foreground application.[222]

Because the court found that all similarities between the plaintiff's and defendant's programs related only to uncopyrightable features, the court dismissed the plaintiff's copyright claim.[223]

---

[219] Id. at 1041.

[220] Id.

[221] Id.

[222] Id. at 1042. These differences mostly seem to relate to functional – and therefore unprotectable – features of the defendant's program, so it is unclear why the court was comparing such features in the comparison step.

[223] Id. The plaintiff had also sought to assert a claim under the Lanham Act based on the alleged similarities between the programs. Without any further analysis, the court simply dismissed this claim, stating in a curious passage the following: "Protection of intellectual property under the Lanham Act does not extend beyond that provided by copyright law, and the

III. ANALYSIS OF THE "FEEL" CASES

The "look" cases focused on copyright infringement claims based primarily on the appearance of various screen display elements of the computer program user interfaces at issue. By contrast, the "feel" cases discussed in this Part III focus on infringement claims based on non-visible or less visible "nonliteral" aspects of computer programs – such as the structure of the code itself, the structure of the menu command system or "menu tree" of the program, and the overall dynamic behavior or flow of the program and the underlying methodologies and program features.

A. THE ASHTON-TATE CASE

The first appellate decision to address the question of whether the menu command structure of a program may be copyrighted was that of Ashton-Tate Corp. v. Ross.[224] That case involved Ashton-Tate's spreadsheet product "Full Impact." In 1984 two programmers, Richard Ross and Randy Wigginton, decided to collaborate on the development of a computer spreadsheet program for the Apple Macintosh computer. Ross wrote the "engine," or computational portion of the product, and Wigginton wrote the user interface. In the course of a "brainstorming" session between the two during the development of the spreadsheet program, Ross gave Wigginton a handwritten list of user commands on a sheet of paper, organized into groups of subcommands, that he thought the program should contain.

In 1985, after developing a prototype of the spreadsheet product, Ross and Wigginton had a falling out and decided to go their separate ways. Wigginton took his user interface code from the prototype to Ashton-Tate, where it was combined with another engine already owned by Ashton-Tate to create the "Full Impact" product. Ross took his engine code from the prototype and wrote new user interface code for it to create a competing spreadsheet product known as "MacCalc."

On the eve of Ashton-Tate's commercial release of Full Impact, Ross asserted that he was entitled to compensation for his "contribution" to the Full Impact program on the ground that, among other things, his handwritten list of commands constituted copyrightable expression which, because it was incorporated into Full Impact, made Ross a joint author of Full Impact.

In response to Ross' claims, Ashton-Tate brought an action for declaratory judgment against Ross to establish that Ashton-Tate owned all existing copyright interest in Full Impact and Ross was not entitled to any compensation from its marketing. The district court held that Ross' handwritten list of commands did not constitute copyrightable authorship, and Ross could therefore not assert joint ownership of Full Impact on the basis that such commands were contained in Full Impact:

---

dismissal of Plaintiff's copyright claims also results in the denial of Plaintiff's motion to add a Lanham Act claim to the Complaint." Id. at 1043.

[224] 916 F.2d 516 (9th Cir. 1990).

> There is nothing innovative or novel about the labels that Ross proposed Wigginton use for the program or the order in which they are listed on the document. The single sheet of paper does not contain any code. ... Ross merely told Wigginton what tasks he believed the interface should allow the user to perform. This list of commands is only an idea that is not protected under federal law.[225]

On appeal, the Ninth Circuit affirmed, holding that Ross' argument that his handwritten list of commands was copyrightable was "meritless for the reasons given in the district court's order.... The list simply does not qualify for copyright protection."[226] The precise reach of this decision is somewhat unclear. Certainly it seems to reject any argument that the mere choice of functions that a program is to perform is not copyrightable expression. However, because the handwritten list of commands also contained a structural ordering of those functions into menus, the case could also be read to stand for the proposition that a program's menu command structure – the hierarchical ordering of commands in the menus of the program – is also not copyrightable, a proposition which the First Circuit agreed with in the case of Lotus Development Corp. v. Borland Int'l,[227] discussed further below.

B. THE ALTAI CASE

One of the most important "feel" cases to be decided to date[228] is the case of Computer Associates Int'l v. Altai Inc.[229] In that case, the court adopted a three-step "abstraction-filtration-comparison test" for judging whether the nonliteral elements of two computer programs are substantially similar. This test, which explicitly rejects the approach of the Whelan[230] decision in judging infringement of nonliteral computer program elements, will, by the court's own admission, probably narrow the scope of such protection.[231] As discussed below, the Second Circuit's abstraction/filtration/comparison structure in the Altai case has been explicitly adopted by the Tenth Circuit and the Fifth Circuit, and has been very influential in other subsequent cases.

---

[225] Ashton-Tate Corp. v. Ross, 728 F. Supp. 597, 602 (N.D. Cal. 1989), aff'd, 916 F.2d 516 (9th Cir. 1990).

[226] 916 F.2d at 521-22.

[227] 49 F.3d 807 (1st Cir. 1995), aff'd by an equally divided court, 116 S. Ct. 804 (1996).

[228] The case has generated considerable controversy. For a severe criticism of the decision, see Clapes & Daniels, "Revenge of the Luddites: A Closer Look at Computer Associates v. Altai," 9 The Computer Lawyer 11 (Nov. 1992).

[229] 982 F.2d 693 (2d Cir. 1992).

[230] Whelan Assocs. v. Jaslow Dental Laboratory, 797 F.2d 1222 (3d Cir. 1986).

[231] "If the test we have outlined results in narrowing the scope of protection, as we expect it will, that result flows from applying, in accordance with Congressional intent, long-standing principles of copyright law to computer programs. Of course, our decision is also informed by our concern that these fundamental principles remain undistorted." 982 F.2d at 712.

1. Background of the Case

The plaintiff in the case was the owner of a computer program called CA-SCHEDULER, which was designed to schedule jobs on an IBM mainframe computer. CA-SCHEDULER contained a subprogram entitled ADAPTER, which was an operating system interface component that enabled a program to run under various operating systems.[232] The defendant Altai also marketed a scheduler program known as ZEKE. In 1984, the defendant hired a former Computer Associates employee who had worked on ADAPTER to rewrite ZEKE. Unknown to the defendant, the employee had taken a copy of the source code to ADAPTER with him when he left Computer Associates.

The employee suggested to Altai that ZEKE be rewritten to contain a "common system interface" component (called OSCAR) similar to the structure of the CA-SCHEDULER/ADAPTER programs. Unknown to Altai, the employee used the source code of the ADAPTER program to create the OSCAR program, and about 30% of the final OSCAR program constituted code taken from the ADAPTER program. When Computer Associates discovered the copying, it brought an action for copyright infringement.

Learning for the first time of the copying, the President of Altai immediately excised those portions of OSCAR that had been copied, and had those portions of OSCAR rewritten by employees who had never had access to the ADAPTER code. Computer Associates maintained in its lawsuit that even the rewritten version of OSCAR infringed Computer Associates' copyrights because of similarities in nonliteral, or structural,[233] elements of the ADAPTER and OSCAR programs. The district court ruled that the rewritten version of OSCAR did not infringe Computer Associates' copyrights.[234]

On appeal, the Second Circuit affirmed.[235] Before beginning its analysis, however, the court cautioned that its decision would not control "categorically distinct works" other than computer code, such as screen displays:

---

[232] CA-SCHEDULER was divided into two components – a first component that contained only task-specific portions of the program, and a second component (ADAPTER) that contained all the interconnections between the first component and the operating system. The first component requested resources from the operating system by making a call to ADAPTER, rather than a system call directly to the operating system. ADAPTER would then translate the call into the appropriate system call to whatever operating system the first component was being run on. Id. at 699.

[233] The court defined a program's "structure" as "its non-literal components such as general flow charts as well as the more specific organization of inter-modular relationships, parameter lists, and macros." Id. at 702. Computer Associates also alleged that OSCAR was substantially similar to ADAPTER "with respect to the list of services that both ADAPTER and OSCAR obtain from their respective operating systems." Id.

[234] Computer Assocs. v. Altai, Inc., 775 F. Supp. 544 (E.D.N.Y. 1991).

[235] Computer Assocs. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).

> These items represent products of computer programs, rather than the programs themselves, and fall under the copyright rubric of audiovisual works. If a computer audiovisual display is copyrighted separately as an audiovisual work, apart from the literary work that generates it (i.e., the program), the display may be protectable regardless of the underlying program's copyright status.[236]

The court noted that the utilitarian nature of computer programs complicates the task of separating idea from expression, particularly in view of the fact that programs combine both "creative and technical expression. ... Thus, compared to aesthetic works, computer programs hover even more closely to the elusive boundary line described in § 102(b)."[237] Because computer programs are made up of various subroutines, and each subroutine may be said to have its own "idea," the court found the Whelan decision's formulation of what constitutes "idea" as a program's overall purpose to be "descriptively inadequate."[238]

2. The Abstraction/Filtration/Comparison Test

The Second Circuit adopted the following three-step formulation for judging substantial similarity of nonliteral elements of computer programs:[239]

1. Abstraction: "in a manner that resembles reverse engineering on a theoretical plane, a court should dissect the allegedly copied program's structure and

---

[236] Id. at 703.

[237] Id. at 704. The court further stated that the "doctrinal starting point in analyses of utilitarian works, is the seminal case of Baker v. Selden, 101 U.S. 99 (1879)." Id. The Selden case held that a set of double-entry bookkeeping "T-account" forms, to the extent they were necessary incidents to use of the non-copyrightable double entry bookkeeping system, could be copied and used by users of the system without copyright infringement. Analogizing to Selden, the Second Circuit concluded that "those elements of a computer program that are necessarily incidental to its function are similarly unprotectable." Id. at 705.

The Second Circuit's reliance on Selden as a fundamental case in the analysis of utilitarian works is in stark contrast to Judge Keeton's approach in the Borland case, in which Judge Keeton rejected Borland's reliance on Selden in arguing that a hierarchy of menu commands constitutes a "system" that is not protectible by copyright. Judge Keeton distinguished Selden as follows: "This is not a case like Baker v. Selden in which the system depends on the use of the copyrighted matter. Borland has, in fact, designed a system (Quattro Pro's native mode), using macros and keystroke sequences and using an alternate command hierarchy, that is fully functional." 799 F. Supp. at 214.

[238] 982 F.2d at 705.

[239] The court cautioned that its test was not fixed in stone. "[I]n cases where the technology in question does not allow for a literal application of the procedure we outline below, our opinion should not be read to foreclose the district courts of our circuit from utilizing a modified version." Id. at 706.

isolate each level of abstraction contained within it. This process begins with the code and ends with an articulation of the program's ultimate function. Along the way, it is necessary essentially to retrace and map each of the designer's steps – in the opposite order in which they were taken during the program's creation [from the top down]."[240]

2. Filtration: "examining the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain and hence is non-protectable expression."[241]

3. Comparison: "Once a court has sifted out all elements of the allegedly infringed program which are 'idea' or are dictated by efficiency or external factors, or taken from the public domain, there may remain a core of protectable expression. ... At this point, the court's substantial similarity inquiry focuses on whether the defendant copied any aspect of this protected expression, as well as an assessment of the copied portion's relative importance with respect to the plaintiff's overall program."[242] Because of the complex and technical nature of computer programs, expert testimony may be used in judging substantial similarity – a strict "lay observer" test need not be applied.[243]

A number of important points should be noted about the Second Circuit's test. This test seems to articulate more clearly than any of the tests used by the courts in the various "look" cases that substantial similarity is to be adjudicated by comparing only that "kernel"[244] of expression in the plaintiff's work that is left after the filtration step is completed. Filtered elements are ignored in the comparison for substantial similarity. As previously discussed, the "look" cases leave largely unanswered the question of what one is to do with elements of a user interface that, standing alone, may be "unprotectable," but may form part of some larger "whole"

---

[240] Id. at 707.

[241] Id. The Second Circuit's filtration step is based upon the "successive filtering method" for separating protectible expression from non-protectible material advocated by Professor Nimmer. See generally 3 M. Nimmer & D. Nimmer, Nimmer on Copyright § 13.03[F].

[242] Id. at 710.

[243] "In making its finding of substantial similarity with respect to computer programs, we believe that the trier of fact need not be limited by the strictures of its own lay perspective. ... Rather, we leave it to the discretion of the district court to decide what extent, if any, expert opinion, regarding the highly technical nature of computer programs, is warranted in a given case." Id. at 713.

[244] "Left with a kernel, or possibly kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program." Id. at 706 (emphasis added).

that is greater than the sum of its individual parts. One may postulate two reasons for this difference in approach between the Second Circuit's test and the "look" cases:

- In cases in which the "look" of a user interface is at issue, one might argue that it is more likely that the totality of visual and interactive elements of the interface may be greater than the sum of its constituent parts, and for this reason the courts may have been content to leave vague the details of precisely what is to be compared at the substantial similarity step. In contrast, when only nonliteral elements of code are at issue (which is all the Second Circuit has said its test is intended to apply to), because these elements are largely functionally designed, one might argue that it is less likely that there will be any larger "whole" that is greater than the sum of the structural parts. If so, it is logical to compare only those elements that are left after sifting out the unprotectable elements of the code's "structure" (as broadly defined by the court).

- In any event, it appears that the "abstraction" and "filtration" steps of the Second Circuit's test allow one to take into account "larger" structural elements that may constitute more than the sum of their parts and therefore constitute expression. The "abstraction" step requires that ideas and possible expressions be defined at all levels of abstraction, and the higher levels of abstraction can take into account "larger" structural elements as possible "expression." If these higher structural levels are found not to be dictated by efficiency, external factors, or drawn from elements in the public domain under the filtration analysis, then such high level program structure could form part of the "kernel" of expression that is compared in the substantial similarity step of the test.

3. Elements That Must Be Filtered Out

Some of the most important aspects of the Altai decision may be found in the elements the court held must be filtered out in the second step of the test:

(a) Elements Dictated by Efficiency

The court noted that "efficiency concerns may so narrow the practical range of choice as to make only one or two forms of expression workable options."[245] Where this is so, those forms must be filtered out of the analysis. Efficiency may dictate choice at high levels of abstraction as well. For example, the court noted that a program's overall modular structure may be dictated by efficiency: "[A] court must inquire 'whether the use of this particular set of modules is necessary efficiently to implement that part of the program's process' being implemented. ... If the answer is yes, then the expression represented by the programmer's

[245] Id. at 708.

choice of a specific module or group of modules has merged with their underlying idea and is unprotected."[246]

(b) Elements Dictated by External Factors

The court noted that the following "extrinsic considerations" may limit a programmer's freedom of design choice in writing a computer program, and may therefore form a basis for limiting the scope of protectable expression in a computer program:[247]

- the mechanical specifications of the computer on which a particular program is intended to run;
- compatibility requirements of other programs with which a program is designed to operate in conjunction;[248]
- computer manufacturers' design standards;
- demands of the industry being serviced; and
- widely accepted programming practices within the computer industry.

(c) Elements Taken From the Public Domain

In this category of unprotectable elements, the court noted "elements of a computer program that have entered the public domain by virtue of freely accessible program exchanges and the like."[249]

4. Application of the Test

The Second Circuit held that the district court's analysis was consistent with its newly articulated three-step test, and therefore affirmed. With respect to the abstraction step, the Second Circuit noted that the district court had postulated five levels of abstraction for purposes of analysis: object code, source code, parameter lists, services required, and general outline.[250]

---

[246] Id. (quoting Englund, "Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs," 99 Mich. L. Rev. 866, 902 (1990)).

[247] 982 F.2d at 709-10.

[248] Although the Second Circuit did not elaborate on this factor, it is interesting to speculate what effect this factor would have in the Borland case, in which Borland argued that the menu command structure of the Lotus 1-2-3 emulation interface of its products was dictated by the need to be compatible with macros written by users of Lotus 1-2-3.

[249] Id. at 710

[250] Id. at 714.

Because the court found no copying of object code or source code in the rewritten version of OSCAR, the court applied its filtration analysis to the parameter lists, services required, and general outline of the ADAPTER program. The district court found that the parameter lists and macros were largely in the public domain or dictated by the functional demands of the program and were therefore unprotectable. The Second Circuit held: "With respect to the few remaining parameter lists and macros, the district court could reasonably conclude that they did not warrant a finding of infringement given their relative contribution to the overall program."[251]

Finally, the district court held that the list of services required by both ADAPTER and OSCAR was determined by the demands of the operating system and of the applications program to which it was to be linked. Accordingly, the Second Circuit concluded that "this aspect of the program's structure was dictated by the nature of other programs with which it was designed to interact and, thus, is not protected by copyright."[252] Because all alleged structural similarities between OSCAR and ADAPTER were unprotectable under the filtration analysis, the Second Circuit affirmed the district court's ruling in favor of the defendant.[253]

C. THE BORLAND CASE

One of the most widely followed "feel" cases was the suit brought by Lotus Development Corp. against Borland International. At the trial court level, this case was assigned to Judge Keeton of the District of Massachusetts, who had earlier decided the Lotus v. Paperback case discussed above.

ANALYSIS OF THE DISTRICT COURT'S DECISIONS

1. Factual Background of the Case

The case involved Borland's spreadsheet products "Quattro" and "Quattro Pro," which Lotus alleged infringed its copyrights in the user interface of Lotus 1-2-3. The Borland products contained their own "native" user interfaces which had a menu command structure or "menu tree"[254] very different from that of Lotus 1-2-3, in function, hierarchical organization and appearance.

---

[251] Id. at 714-15.

[252] Id. at 715.

[253] Computer Associates also alleged that certain similarities in "organizational charts" between the two programs were evidence of copying. The district court accorded no weight to these similarities "because [the charts were] so simple and obvious to anyone exposed to the operation of the program[s]," and the Second Circuit agreed. Id. The Second Circuit also rejected Computer Associates' challenges to several of the district court's factual findings regarding the creative nature of its program components, ruling that such findings were not clearly erroneous.

[254] "'Command structure' refers to the organization of the menus and menu commands. (Other phrases used with essentially the same meaning include 'menu command structure,' 'menu hierarchy,' and 'menu command hierarchy.') In Lotus 1-2-3, menu commands are organized

Until 1992,[255] the Borland products also contained an optional "emulation interface" that the user could call up to the screen, which contained all of the commands of Lotus 1-2-3, with a great many other commands not contained in Lotus 1-2-3 interspersed throughout the menu tree. The Lotus 1-2-3 commands were, however, presented visually to the user in the "look" of the Borland native interface, which contained pop up menus (in the case of Quattro) and pull down menus, icons, buttons and other graphical elements (in the case of Quattro Pro). Users of Lotus 1-2-3 could use the emulation interface either to execute macros written for Lotus 1-2-3, or to use the familiar commands of Lotus 1-2-3. Thus, the Borland products, when used with the emulation interface, would "feel" the same to a user as Lotus 1-2-3 – in the sense that the same commands and keystrokes could be used to accomplish the same functions as in Lotus 1-2-3 – but the programs did not "look" similar.[256]

Lotus brought a copyright infringement lawsuit against Borland, alleging that the emulation interfaces of the Borland products (although not the "native" interfaces) infringed Lotus' copyrights in the user interface of Lotus 1-2-3. Judge Keeton ultimately ruled in favor of Lotus, after issuing a series of four lengthy opinions. Those opinions are each discussed in turn below. The case was appealed by Borland to the First Circuit,[257] which reversed on the ground

---

so that less than a dozen related menu commands are displayed at any given moment. ... Each menu of less than a dozen commands is linked to preceding/succeeding menus by the operation of menu commands. All command menus are ultimately linked to a single main (root/trunk) menu to form a 'menu tree.'" Lotus Development Corp. v. Borland Int'l, 799 F. Supp. 203, 206 (D. Mass. 1992).

[255] Borland announced in August of 1992 that it would remove the 1-2-3 "emulation" interface from its spreadsheet products. See Wilke, "Ruling Against Borland May Intensify Copyright Debate," The Wall Street Journal, Aug. 3, 1992, at B1.

[256] As Judge Keeton noted: "In this case, Borland has appropriated, to a great extent, the 'feel' of the 1-2-3 user interface and only to a lesser extent the 'look' of 1-2-3. Indeed, Borland has designed an interface that in many respects looks substantially different from the 1-2-3 user interface. ... The 'feel,' on the other hand, of the emulation modes of the Quattro programs depends in large part on the keystroke sequences one enters to perform spreadsheet operations. One enters the same keystroke sequence to perform the same spreadsheet operations in both 1-2-3 and Quattro Pro's emulation mode. They feel the same." 799 F. Supp. at 220.

[257] Judge Keeton's decisions were widely criticized. An amicus brief was filed in the First Circuit signed by 24 professors of copyright law from various law schools around the country, arguing that Judge Keeton erred in the legal test he formulated for judging what elements of a computer program are copyrightable and in applying the proscriptions deriving from Baker v. Selden, 101 U.S. 99 (1879), against protecting systems, methods and the like under copyright. Other amicus briefs filed in support of Borland's position on appeal were filed jointly by Professors Dennis Karjala and Peter Menell (both professors of copyright law), and by the American Committee for Interoperable Systems (ACIS), by a group of computer scientists, by the Software Entrepreneurs' Forum, and jointly by various PC user groups. See Computer Industry Litigation Reporter, Jan. 6, 1994, at 17957 (Andrews Publications).

that the Lotus 1-2-3 menu structure is not copyrightable under § 102(b) of the copyright statute. The Supreme Court ultimately affirmed the First Circuit's decision by an equally divided court.

2. Borland I

Judge Keeton's first decision in the case set the analytical framework for all later decisions. In the earlier Paperback case, Judge Keeton had concluded that the user interface of Lotus 1-2-3 taken as a whole was copyrightable. The court noted the following, however, in the first decision issued in March of 1992: "The conclusion that the user interface as a whole is copyrightable (which this court reached in Paperback) does not resolve the further questions that may now have to be resolved regarding the copying and copyrightability of individual parts or a sum of parts less than the user interface as a whole."[258]

The court stated that Lotus would have to establish three fundamental elements to prove infringement by Borland:

> (a) identify expressive elements in 1-2-3 that were indisputably copied in the Quattro programs, (b) establish that those expressive elements, either separately or together, are as a matter of law copyrightable, and (c) establish that the copied expressive elements of the Quattro programs' emulation interfaces are substantially similar to copyrightable elements of the 1-2-3 interface.[259]

---

[258] Lotus Development Corp. v. Borland Int'l, 788 F. Supp. 78, 81 (D. Mass. 1992) ("Borland I").

[259] Id. In establishing these three elements, "substantial similarity" may actually be used twice. It may first be used in proving "copying" by the defendant under element (a):

> Copying can, of course, be proved directly, and there is some direct evidence of copying in this case. However, once access is proved, copying can also be proved by demonstrating "substantial similarity." In this context, "substantial similarity" simply means sufficient similarity of a given element of a work to an element in the allegedly infringing work to support a reasoned inference that more probably than not the element was copied from the copyrighted work. This is not similarity in a mixed law-fact sense that includes being similar enough to constitute "unlawful appropriation." Rather, the elements of the copyrighted and allegedly infringing work must be shown to be substantially (i.e., notably) similar in a purely factual sense. This is "substantial similarity" in an evidentiary sense. "Substantial similarity" in this sense is one kind of circumstantial evidence of copying.

Id. at 84. "Substantial similarity" is then used a second time under element (c) to establish unlawful appropriation of expression that has been ruled copyrightable under element (b). This second use of "substantial similarity" is not evidentiary, but rather is used in what Judge Keeton calls "a mixed law-fact sense" to establish sufficient similarity to conclude "unlawful appropriation" of protected expression. Id. In the first use of substantial similarity, similarities in both protectable and unprotectable elements may be considered. In the second use of substantial similarity, only similarities in protectable elements may be considered.

With respect to element (b), the court adopted the same basic three-step test for copyrightability of an allegedly copied element as it had used in the Paperback case,[260] except that, in response to criticism contained in an amicus brief submitted by a number of copyright law professors urging that the Paperback opinion "gave inadequate attention and emphasis to the distinction between a copyrightable expression and a useful process,"[261] Judge Keeton revised his test by simply replacing the phrase "idea" at each place at which it appeared in his Paperback test, with the phrase "idea," "system," "process," "procedures," or "method." The court concluded that this gloss on the Paperback test for copyrightability was desirable to recognize in a more explicit way that systems, processes, procedures and methods are, like ideas, unprotectable under copyright law. The court also concluded that the issue of copyrightability, including any fact questions bearing upon it, must be determined by the court, not the jury.[262]

3. Borland II

The court rendered its second decision in July of 1992. The court found element (a) (copying) of Lotus' infringement case established because it found that "Borland has admitted that it intentionally incorporated into its user interface the 1-2-3 menu commands and menu command hierarchy."[263] The court further found that "Borland admits to copying the functionality of the keystroke sequences and macro language," but concluded that it was an issue for the jury whether Borland copied the long prompts of Lotus 1-2-3.[264] Having concluded that Borland copied the menu commands, menu command hierarchy, keystroke sequence, and macro language of Lotus 1-2-3, the court proceeded to "determine whether those aspects of the 1-2-3 user interface, taken together, are copyrightable"[265] under step (b) of the infringement test.

Not surprisingly in view of the court's decision in Paperback, Judge Keeton ruled that the elements Borland had copied from Lotus 1-2-3 were indeed copyrightable. Applying the first step of the court's modified Paperback test for copyrightability – formulating the various ways in which the "idea" underlying Lotus 1-2-3 might be defined at various levels of abstraction – the court laid out a number of definitions that might be used, from the most abstract ("an electronic

---

Protectable elements may, however, be comprised of combinations of individually unprotectable elements. Id. at 81-82, 84.

[260] See supra pp. 10-11.

[261] 788 F. Supp. at 89-90.

[262] Id. at 96.

[263] Lotus Development Corp. v. Borland Int'l, 799 F. Supp. 203, 208 (D. Mass. 1992) ("Borland II").

[264] Id. at 209.

[265] Id. (emphasis added).

spreadsheet"[266] or "a menu-driven electronic spreadsheet") to the most particular (a spreadsheet having an interface containing the precise set of menu commands in Lotus 1-2-3).[267]

The court chose the following definition, which was one level of abstraction above the most particular one it had posited:

> Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy.[268]

The court was thus willing to define the "idea" underlying Lotus 1-2-3 as a spreadsheet having the same functions as 1-2-3 organized hierarchically into a menu tree, though not necessarily having the same particular command names and hierarchical organization as 1-2-3. The court concluded that "the selection of functional operations that the spreadsheet performs must be considered part of the idea of the program. Copyrightability depends on expression distinct from the selection of the set of spreadsheet operations that can be performed."[269]

Applying the second step of the test – whether there exists identifiable elements of expression not essential to every expression of that idea – the court concluded that a satisfactory spreadsheet menu tree could be constructed using different commands and a different hierarchy from that of Lotus 1-2-3. Indeed, Borland had constructed one such menu tree in its "native" mode interfaces. Lotus' particular menu tree was therefore not dictated by functional considerations, and thus not essential to every expression of the idea.[270] Similarly, the court found that "it was not necessary to copy expressive aspects of the macro language and keystroke sequences to copy their function."[271] Other keystroke sequences accomplishing the same functions could have been used.

Finally, applying the third step of the test – whether the allegedly copied expression formed a substantial part of the allegedly copyrightable work – Judge Keeton ruled that no reasonable jury could not find that "the creativity involved in establishing the menu commands, menu command hierarchy, macro language, and keystroke sequences was more than trivial."[272]

---

[266] The court noted that a definition of the "idea" at this highest level of abstraction would be consistent with the approach taken by the Whelan court, but rejected the Whelan decision's approach as too broad. Id. at 217.

[267] Id. at 216.

[268] Id.

[269] Id. at 217

[270] Id. at 217-19.

[271] Id. at 219 (emphasis in original).

[272] Id.

Under this low threshold of creativity, Lotus had satisfied all three prongs of the test, and Judge Keeton concluded that the elements of the Lotus 1-2-3 interface Borland had copied were copyrightable, and Borland therefore infringed by copying the menu command hierarchy, keystrokes and macro language of 1-2-3.[273]

Despite Judge Keeton's conclusion that no reasonable factfinder could find that Borland had not illicitly copied copyrightable expression from Lotus, his opinion contains confusing language on the issue of how one should treat uncopyrightable elements in judging substantial similarity. In Borland I, Judge Keeton stated that "'[s]ubstantial similarity' ... indicates a degree of similarity between the allegedly infringing material and what is copyrightable (that is, the copyrightable part or parts)."[274] In Borland II, however, he stated, "Substantial similarity (in the mixed law-fact sense) is determined by comparing the copied copyrightable elements of the infringed work all together with the copyrighted work as a whole."[275] This suggests a comparison against the copyrighted work as a whole, including the uncopyrightable elements. Despite this confusion of language, however, it seems that Judge Keeton concluded that in the case at hand there was sufficient similarity in "feel" between the two works – with or without consideration of the other uncopyrightable elements – for a finding of infringement as a matter of law.[276]

In another important part of the opinion, Judge Keeton rejected Borland's argument against copyrightability based on requirements of compatibility. Borland asserted that the 1-2-3 interface was not copyrightable because the menu command hierarchy "was dictated by the

---

[273] "Even if I assume ... that Borland did not copy the long prompts, and that some aspects of the menu commands, menu command hierarchy, macro language, and keystroke sequences of 1-2-3 are not copyrightable, I conclude that no reasonable jury, applying the law, could find other than that the Quattro programs infringe 1-2-3. That is, a reasonable factfinder must conclude that the Quattro programs derive from illicit copying. The emulation interfaces are substantially similar in the mixed law-fact sense to the Lotus 1-2-3 user interface. (Returning to the metaphor, one may say that is why they 'feel' the same.)" Id. at 221. Judge Keeton left open, however, the question "whether Borland is prohibited from reading and interpreting the macros that have been created by users of 1-2-3." Id. at 214.

[274] 788 F. Supp. at 84 (emphasis added).

[275] 799 F. Supp. at 221 (emphasis added).

[276] Borland argued that its placement of the Lotus 1-2-3 menu hierarchy within the "look" of its "native" interface, and the addition of a great many other commands to the hierarchy not found in Lotus 1-2-3, had so transformed the copied elements that there was no substantial similarity between the Lotus 1-2-3 interface and the Borland emulation interfaces overall. Judge Keeton, in a passing remark, probably unwittingly lent credence to this argument, although he gave no specific treatment to Borland's argument: "A decisionmaker in this case (whether judge or jury) must ignore the ... expression [added by Borland] to the extent that it does not change the expression Borland copied from Lotus." Id. at 222 (emphasis in original). One may infer from the outcome of the case, however, that Judge Keeton did not find the added expression of Borland resulting from its "native" look and feel to be sufficient to "change the expression Borland copied from Lotus."

nature of the user macros with which it was designed to interact."[277] The court ruled that this argument was flawed because it implicitly assumed that the macros which Lotus 1-2-3 was designed to run were in existence before the menu hierarchy of 1-2-3 was designed.

> [A] program designed to interact with preexisting software ... is not entitled to protection to the extent that it is constrained by the need for compatibility with the preexisting software. ... The Lotus 1-2-3 interface – or at least a version of it – was written first. All user macros derive from it. Thus, Borland is simply wrong factually to argue that the 1-2-3 interface was constrained by the macros.[278]

Thus, Judge Keeton was willing to recognize "compatibility" with programs written by others as an "externality" that would limit the scope of copyright protection only if such other programs preexisted the copyrighted work at issue. This view, carried to its logical conclusion, will effectively preclude any existing base of user macros or files that were written to conform to a particular program (such as database files or other data files in a format dictated by the language or menu structure of a program that created them) from ever becoming an "externality" that could justify a third party copying the elements of a preexisting program necessary to offer a competing program that is compatible with such existing base. It is, therefore, a narrow view of when "compatibility" with an existing standard can justify copying of what might otherwise be deemed copyrightable expression.[279]

4. Borland III

Although Judge Keeton determined that Borland had done sufficient copying from Lotus 1-2-3 to establish liability on Borland's part as a matter of law, the Borland II decision left open certain factual issues to be determined at trial concerning the scope of Borland's impermissible copying. Specifically, as reported in the Borland III decision, the Borland II decision left open specific fact issues relating to "(1) whether Borland copied the long prompts of Lotus 1-2-3, (2) whether the long prompts contain expressive elements, and (3) the extent (if any) that functional constraints limit the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation."[280]

Following the Borland II decision, the parties stipulated that the remainder of the liability issues were to be tried to the court, rather than to a jury, including trial of Borland's defenses of laches and estoppel, as well as its affirmative defense of fair use, which had been newly raised by Borland at the close of its evidence in the trial. The parties also stipulated that neither would contend, either in the case or in any appeal therefrom, that Borland had or had not copied the

---

[277] Id. at 212.

[278] Id. at 213.

[279] Cf. Paperback, 740 F. Supp. at 69: "[T]he desire to achieve 'compatibility' or 'standardization' cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual 'work.'"

[280] Lotus Development Corp. v. Borland Int'l, 831 F. Supp. 202, 207 (D. Mass. 1993) ("Borland III").

long prompts of Lotus 1-2-3, or that the issue of copying of long prompts was material to any other issue in the case.[281] This stipulation removed items (1) and (2) above from the trial. Accordingly, the only issues before the court at trial relating to the scope of infringement concerned item (3) – the extent to which functional constraints limit the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation.

Borland alleged that eight functional considerations limited the design of the 1-2-3 menu command hierarchy:

1) Each command was chosen to tell the user its purpose and function.

2) Each command was selected so that it had a different first letter from other commands in the same menu.

3) Each menu was set up to have only seven choices, plus or minus two.

4) Menus were structured so that similar command functions were grouped together.

5) Executable operations likely to be frequently used were located near the top of the command hierarchy.

6) Menu commands within a menu were arranged from left to right in order of decreasing frequency of use.

7) Commands in submenus were grouped under the menu command to which they relate.

8) Each menu could have no more than 80 characters so as to fit on one line on the screen.[282]

Judge Keeton ruled that these functional considerations did not significantly constrain the number of menu hierarchies available to Lotus or Borland to essentially only one form, even if one were setting out to provide the same functions as Lotus 1-2-3. The court noted that, because the Lotus 1-2-3 menu hierarchy contains 469 commands, even if there were only two acceptable words available for each individual command, there would theoretically be 2 raised to the 469th power possible menu trees available based on word choices alone. When one adds possible choices in structure, the total range of possibilities becomes even larger. The court also noted the existence of vastly different menu trees in other commercial programs as proof that the Lotus 1-2-3 menu tree was but one of many possible forms for a menu tree.[283]

---

[281] Id. at 208.

[282] Id. at 212-13.

[283] Id. at 213-14.

Accordingly, the court rejected Borland's argument that copying of the Lotus 1-2-3 menu hierarchy was permissible because of functional constraints on the formulation of the menu commands and structure. For similar reasons, the court also rejected Borland's argument that, because of the functional considerations used to design the Lotus 1-2-3 menu hierarchy, it did not possess sufficient originality to be copyrightable.[284]

5. Borland IV (The "Key Reader" Decision)

The fourth and final decision issued by Judge Keeton involved adjudication of additional allegations raised by Lotus in a supplemental complaint it filed against Borland in January 1992, alleging copyright infringement based on the "Key Reader" feature in Borland's spreadsheet programs. With the Key Reader feature turned on, when Borland's programs encountered a slash key ("/") in a macro, the program would interpret everything that followed the slash as though it were part of a macro written for use with Lotus 1-2-3. This feature enabled Borland's programs to execute Lotus 1-2-3 macros without displaying the Lotus 1-2-3 menu commands on the screen.[285]

Judge Keeton determined that the Key Reader feature also infringed Lotus' copyrights in 1-2-3, based largely upon how that feature was implemented through certain data files stored in the Borland programs. Accordingly, it is necessary to set forth some of the technical details about how both the emulation interfaces and the Key Reader features were implemented in Borland's products.

(a) Technical Details of How the Key Reader Feature Was Implemented

The actual menu tree for the emulation interfaces was stored in a file used by the programs (the file was labeled "123.MU" in Quattro Pro, for example). When the emulation interface was selected by the user, the program would refer to the 123.MU file to determine the form of menu commands and structure to present to the user on the screen, to interpret user commands, and to interpret macros. If the user selected a different interface, the program would refer to a different file for display and interpretation of commands, and for macro execution.[286] Judge Keeton found that the 123.MU file contained a copy of the entire Lotus 1-2-3 menu command hierarchy: "[T]he entire Lotus menu tree is copied into the file, with differences in

---

[284] Id. at 215-17. The court also rejected Borland's defenses of laches and estoppel for reasons beyond the scope of this article.

[285] Lotus Development Corp. v. Borland Int'l, 831 F. Supp. 223, 226-27 (D. Mass. 1993) ("Borland IV"). "Borland removed the emulation interface from Quattro Pro version 4.01 (and subsequent releases of Quattro Pro) after the Borland II decision in this case allowed partial summary judgment for Lotus. Thus, Quattro Pro versions 4.01, SE, and Quattro Pro for Windows contain the Key Reader feature but have no 1-2-3 emulation interfaces." Id. at 227.

[286] Id. at 228.

indentation detailing the menu structure. That is, the structure of the menus and submenus is recorded in the file by changes in indentation."[287]

To implement the Key Reader feature, the court found that Borland began with the 123.MU file from the emulation interface and "prepared a new file [which Judge Keeton referred to as the "Key Reader file"] by reproducing the old 123.MU file but with only the first letter of each menu command name where the entire Lotus menu command name appears in the old 123.MU file. Put another way, the point is that to implement Key Reader Borland used a program file containing the same copy of the 1-2-3 menu structure and commands that Borland had used in its emulation interface, but with each menu command name stripped of everything after the first letter."[288] The menus embodied in the Key Reader file were sometimes referred to by the parties as "phantom menus," because they were used in interpreting macros but never fully displayed to the user.[289]

Based on the content and structure of the Key Reader file, the court concluded "that the Key Reader file contains a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names."[290] Because the court had earlier held that the Lotus menu tree structure was copyrightable, and because he found that the Key Reader file contained a virtually identical copy of that structure, Judge Keeton ruled that the Key Reader feature infringed Lotus' copyrights.

(b) "On-the-Fly" Versus "One-Time" Translation

In the course of attempting to clarify the basis for his ruling, Judge Keeton drew a technical distinction between "on-the-fly" and "one-time" translation of macros that renders unclear the scope of his holding with respect to what forms of macro interpretation may constitute copyright infringement. Specifically, he used the term "on-the-fly" translation to refer to Borland's interpretation of a macro written in Lotus 1-2-3 macro language as it is executed, using the Key Reader file. Under the "on-the-fly" technique, each time a macro is executed,

---

287 Id.

288 Id. (citation omitted).

289 Id.

290 Id. The court found irrelevant Borland's contention that the command letters copied from Lotus 1-2-3 in the Key Reader file, if read sequentially down the file, were not in the same order as displayed on the screen in Lotus 1-2-3: "This is true, however, only in the sense that the menu structure of Lotus 1-2-3 is represented in a different way in the Key Reader file than on the Lotus 1-2-3 display screens; in the file, the structure is detailed by differences in indentation (or other means) rather than through display on the screen. I find that the file ... fully delineates a virtually identical copy of the menu structure of Lotus 1-2-3 including the first letter of each menu command in the corresponding location in the copy of the menu structure." Id. at 228-29.

modified or debugged, the program refers to the "phantom menus" in the Key Reader file. The macro remains, however, in the "native" macro language in which it was originally written.[291]

By contrast, Judge Keeton used the term "one-time" translation to refer to the technique of translating a macro from one macro language to another macro language (for use with a different menu tree), then executing the macro in its translated form. Because the translated macro is written in a different macro language, the program need not refer to a Key Reader file to execute the translated macro.[292]

Judge Keeton ruled that Borland's "on-the-fly" macro interpretation, because it relied on the Key Reader file which he found contained a copy of the Lotus 1-2-3 menu structure, constituted infringement. Lotus had argued that "on-the-fly" macro interpretation does not require copying from the Lotus menu structure and first letters of the command names, and that Borland therefore could have implemented "on-the-fly" interpretation without such copying. Interestingly, Judge Keeton rejected Lotus' argument, finding that "[t]o interpret a macro, the program must use the Lotus 1-2-3 menu structure."[293] Nevertheless, he ruled that when such menu structure is used, as he believed it must be, it constitutes infringement. He explicitly declined in Borland IV, however, to decide "whether copying of the Lotus menu structure for the purpose of one-time translation rather than on-the-fly interpretation should be accorded different treatment under copyright law."[294]

It is unclear precisely what Judge Keeton means by his statement that "on-the-fly" macro interpretation must "use" the Lotus 1-2-3 menu structure. Lotus may in fact have been correct in its argument that one could implement an "on-the-fly" interpreter without literally copying the Lotus menu structure and first letters of the command names. One can imagine, for example, an implementation of "on-the-fly" interpretation that does not require a Key Reader file. This could be accomplished simply by viewing the Lotus menu tree as a decision tree, and implementing the structural relationships embodied in that decision tree in a series of in-line "If-Then-Else" statements in the code of the macro interpreter.

Although such an implementation would not require a literal copy of the menu tree in a Key Reader file, it would certainly "use" the Lotus 1-2-3 menu structure in the sense of implementing that structure in the logic of the program flow as represented by the "If-Then-Else" statements. Although it is doubtful whether Judge Keeton understood this technical alternative for "on-the-fly" interpretation, there is nevertheless language in his opinion that could be interpreted to mean that such an implementation would also be infringing. Specifically, Judge Keeton stated his belief that there is no way to interpret a macro "[i]f a program did not have a

---

[291] Id. at 229.

[292] Id. Judge Keeton did not address whether it would be necessary to reference a Key Reader file during the "one-time" translation process itself. If so, however, even the "one-time" translation process might require a copy of the Lotus 1-2-3 menu structure in the form of a Key Reader file.

[293] Id. at 230.

[294] Id.

representation of the 1-2-3 menu hierarchy somewhere within the program code (or in a file that is used by the code)".[295] If an implementation of the Lotus 1-2-3 menu hierarchy as a decision tree in a series of "If-Then-Else" statements is viewed as a representation of the 1-2-3 menu hierarchy "somewhere within the program code," then even this implementation would be infringing under Judge Keeton's analysis. If so interpreted, then the effect of Judge Keeton's ruling may be to make all forms of "on-the-fly" macro interpretation infringing, regardless of how implemented.

(c) Rejection of Borland's Arguments

Borland contended that copying of the 1-2-3 menu structure and first letters of command names in the Key Reader file was a necessary part of any "system" for interpreting Lotus 1-2-3 macros, and that Lotus' copyrights could therefore not extend to Borland's phantom menus. Judge Keeton rejected this argument virtually out of hand:

> Borland wishes the court to define the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 as including the ability to interpret macros written for use with Lotus 1-2-3 ... and I reject this argument .... The fact that users of Lotus 1-2-3 have created macros in reliance on expressive aspects of Lotus 1-2-3 does not convert that expression into a part of the "system." That Borland wishes to copy protected expression contained in Lotus's menu tree for what Borland contends is a utilitarian purpose also does not turn that expression into a "system" under copyright law.[296]

In short, Judge Keeton was unwilling to view the Lotus menu structure as a system for invoking the functions in Lotus 1-2-3 by virtue of the fact that such structure was necessary to run macros that depended upon such structure. Instead, he insisted on coming at the issue from the other direction – first defining the structure as "expression," then holding that one may not copy such expression even as part of the implementation of what might be characterized as a "system" for invoking the functions of 1-2-3.[297]

Judge Keeton also rejected an argument that the phantom menus were part of a "system" because they were not displayed on the screen by the products. Analogizing to computer code, which he stated is protectable by copyright but may never be seen by the user, he found irrelevant the fact that the phantom menus were not displayed. He also found that such menus could, in any event, be printed out in the form of the Key Reader file.[298]

---

[295] Id. (emphasis added)

[296] Id. at 231-32.

[297] Id.

[298] Id. at 232. Judge Keeton also found that the phantom menus in Borland's products were substantially similar to the Lotus 1-2-3 menu structure:

> I also conclude that differences in the method Borland uses to represent the menu tree structure in its phantom menus file does not negate a finding that the copied expression of the menu tree structure is substantially similar to the Lotus 1-2-3

ANALYSIS OF THE FIRST CIRCUIT'S DECISION

On appeal, the First Circuit reversed Judge Keeton's decisions.[299]

1. Preliminary Analysis

The Court began its legal analysis by noting that in the First Circuit, copyright infringement is established under a two prong test of (i) proof of factual copying, followed by (ii) proof of illicit copying. Factual copying is established by "direct evidence of factual copying or, if that is unavailable, evidence that the alleged infringer had access to the copyrighted work and that the offending and copyrighted works are so similar that the court may infer that there was factual copying (i.e., probative similarity)."[300] Illicit copying is established by proving "that the copying of copyrighted material was so extensive that it rendered the offending and copyrighted works substantially similar."[301] Thus, similarity is used twice in this test for infringement.

The Court noted, however, that the issue of whether copying took place need not be analyzed at all if what was allegedly copied does not constitute copyrightable subject matter. Accordingly, the Court turned to the threshold question (and central issue on appeal) of "[w]hether a computer menu command hierarchy constitutes copyrightable subject matter."[302] The First Circuit initially looked to whether existing precedent among the computer software copyright cases would assist in deciding this legal issue. Although observing that some other courts (notably the Tenth Circuit in the Autoskill case discussed below) had touched on the issue in dicta, the Court stated that "we know of no cases that deal with the copyrightability of a menu command hierarchy standing on its own (i.e., without other elements of the user interface, such as screen displays, in issue)."[303]

> program. First, copyright law protects nonliteral aspects of a copyrighted program. Thus, one need not copy the specific code of a program to infringe copyrights in the program. Second, Borland's copying is analogous to a translation. ... Here, Borland created a virtually identical copy of the Lotus menu structure, but translated (nearly verbatim) the menu structure into a different language for representing menu structures.

Id. at 234. Judge Keeton also rejected Borland's fair use defense to copying of the phantom menus, for reasons that are beyond the scope of this Article. See id. at 240-45.

299 Lotus Development Corp. v. Borland Int'l, 49 F.3d 807 (1st Cir. 1995), aff'd by an equally divided court, 116 S. Ct. 804 (1996).

300 Id. at 813.

301 Id.

302 Id.

303 Id.

The Court determined that the Altai case and its abstraction/filtration/comparison test would not be helpful in resolving the issue before it for two reasons. *First*, the Court noted that the Altai test was designed to deal with an allegation of copying of code – specifically, "whether one computer program copied nonliteral expression from another program's code."[304] The Court noted that it was not in the present case faced with an issue of copying of nonliteral elements of computer program code, but rather of literal copying of a menu command hierarchy, and the test would therefore not be helpful.

Second, the Court noted that

> the Altai test in this context may actually be misleading because, in instructing courts to abstract the various levels, it seems to encourage them to find a base level that includes copyrightable subject matter that, if literally copied, would make the copier liable for copyright infringement. ... We think that abstracting menu command hierarchies down to their individual word and menu levels and then filtering idea from expression at that stage, as both the Altai and the district court tests require, obscures the more fundamental question of whether a menu command hierarchy can be copyrighted at all. The initial inquiry should not be whether individual components of a menu command hierarchy are expressive, but rather whether the menu command hierarchy as a whole can be copyrighted.[305]

2. The Menu Command Hierarchy as a "Method of Operation"

Having found no existing precedent useful in resolving the issue before it, the First Circuit turned to an analysis from first principles of the issue of whether a computer menu command hierarchy constitutes copyrightable subject matter. The Court did not adopt Judge Keeton's three-step test for copyrightability of an allegedly copied element, but turned instead directly to the limitations on copyrightable subject matter contained in § 102(b).[306] The Court concluded that the Lotus menu command hierarchy constituted an uncopyrightable "method of operation" under § 102(b).

The Court defined a "method of operation" to mean "the means by which a person operates something, whether it be a car, a food processor, or a computer," and concluded that the Lotus menu command hierarchy fell within this definition because it "provides the means by which users control and operate Lotus 1-2-3."[307]

In arguing that its menu command hierarchy constituted copyrightable subject matter, Lotus had relied heavily in the district court on the argument that the hierarchy contained expression because it communicated to the user the choices available to accomplish spreadsheet tasks. The First Circuit rejected this argument, finding that the words were not merely

---

[304] Id. at 814.

[305] Id. at 815.

[306] 17 U.S.C. §102(b).

[307] 49 F.3d at 815.

expressive: "The Lotus menu command hierarchy does not merely explain and present Lotus 1-2-3's functional capabilities to the user; it also serves as the method by which the program is operated and controlled."[308] The Court noted that, unlike the program's underlying code, which it was not necessary to copy in order to have a program operate the same way as Lotus 1-2-3, "to allow users to operate its programs in substantially the same way ... Borland had to copy the Lotus menu command hierarchy."[309]

The First Circuit found that the fact that the Lotus menu command hierarchy "serves as the basis for Lotus 1-2-3 macros"[310] bolstered its conclusion that the menu command hierarchy constitutes an uncopyrightable method of operation: "That programs can offer users the ability to write macros in many different ways does not change the fact that, once written, the macro allows the user to perform an operation automatically."[311]

In one of the most significant conceptual portions of its opinion, the First Circuit rejected the district court's conclusion that merely because the Lotus developers made some "expressive" choices in choosing and arranging the Lotus command terms, the resulting menu command hierarchy was therefore copyrightable expression. Instead, the Court held that words themselves can constitute a method of operation where such words are the direct mechanism for invoking operations:

> We do not think that "methods of operations" are limited to abstractions; rather, they are the means by which a user operates something. If specific words are essential to operating something, then they are part of a "method of operation" and, as such, are unprotectable. This is so whether they must be highlighted, typed in, or even spoken, as computer programs no doubt will soon be controlled by spoken words.[312]

Moreover, in another highly significant portion of its analysis, the Court rejected the district court's fundamental premise that, because other menu command hierarchies could have

---

[308] Id. By contrast, the Court held that the long prompts, the possible copying of which was not at issue on appeal, could potentially constitute protectable expression "for the long prompts are not necessary to the operation of the program; users could operate Lotus 1-2-3 even if there were no long prompts." Id. The Court noted in dicta, however, that a strong argument could be made that the brief explanations the long prompts provide merge with their underlying idea. Id. at 815 n.9. Similarly, the Court noted that the Lotus screen displays were potentially copyrightable expression, because "users need not 'use' any expressive aspects of the screen displays in order to operate Lotus 1-2-3; because the way the screens look has little bearing on how users control the program, the screen displays are not part of Lotus 1-2-3's 'method of operation.'" Id. at 816.

[309] Id. at 816.

[310] Id. at 818.

[311] Id.

[312] Id. at 816.

been designed to accomplish the same spreadsheet functions, Lotus' particular menu command hierarchy was therefore copyrightable:

> Concluding, as we do, that users operate Lotus 1-2-3 by using the Lotus menu command hierarchy, and that the entire Lotus menu command hierarchy is essential to operating Lotus 1-2-3, we do not inquire further whether that method of operation could have been designed differently. The "expressive" choices of what to name the command terms and how to arrange them do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter.[313]

The Court analogized the Lotus menu command hierarchy to the buttons used to control a video cassette recorder (VCR). "That the buttons are arranged and labeled does not make them a 'literary work,' nor does it make them an 'expression' of the abstract 'method of operating' a VCR via a set of labeled buttons. Instead, the buttons are themselves the 'method of operating' the VCR."[314] The Court analogized the choosing of a command from the Lotus 1-2-3 menu command hierarchy, either by highlighting it on the screen or by typing its first letter, to pushing a button:

> Just as one could not operate a buttonless VCR, it would be impossible to operate Lotus 1-2-3 without employing its menu command hierarchy. Thus the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves. ... Without the menu commands, there would be no way to "push" the Lotus buttons, as one could push unlabeled VCR buttons.[315]

The Court noted that its holding that methods of operation are not limited to abstractions was contrary to the Tenth Circuit's decision in the Autoskill case, discussed in the next section, in which the Tenth Circuit rejected the defendant's argument that the keying procedure used in a computer program was an uncopyrightable "procedure" or "method of operation" under §

---

[313] Id.

[314] Id. at 817.

[315] Id. The Court noted that one might argue that buttons for operating a VCR are not analogous to computer commands because VCRs are not copyrightable, whereas computer programs are. In particular, the buttons on a VCR would be subject to the useful article doctrine of copyright law, which protects only designs that incorporate pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article. The Court noted that unlike VCRs, computer programs are protected as literary works. Accordingly, one might argue that the "buttons" used to operate a computer program are not like the buttons used to operate a VCR, for they are not subject to the useful article doctrine. "The response, of course, is that the arrangement of buttons on a VCR would not be copyrightable even without a useful-article exception, because the buttons are an uncopyrightable 'method of operation.' Similarly, the 'buttons' of a computer program are also an uncopyrightable 'method of operation.'" Id.

102(b). The Tenth Circuit found the keying procedure used by the plaintiff's program, in which the user selected responses to the program's queries by pressing the 1, 2, or 3 keys, to reflect at least the minimal degree of creativity required under copyright law. The First Circuit disagreed with this view:

> As an initial matter, we question whether a programmer's decision to have users select a response by pressing the 1, 2, or 3 keys is original. More importantly, however, we fail to see how "a student select[ing] a response by pressing the 1, 2, or 3 key" can be anything but an unprotectable method of operation.[316]

3. Importance of the Decision

Judge Keeton's decisions in the Borland case, if the First Circuit had affirmed them on appeal, would have had far reaching effects on cross-product compatibility that is dependent upon the command structure of a program. Judge Keeton's opinions could have been interpreted to prohibit compatibility with the menu command structure of another's program, regardless of how such compatibility is implemented, and regardless of whether the commands are displayed on the screen.

By contrast, the logic of the First Circuit's decision calls into question whether any computer program menu command structure can be copyrightable, since any such command structure could be described as "the means by which users control and operate" the program, and therefore be deemed a "method of operation" within the First Circuit's definition of that term. Indeed, the First Circuit explicitly noted the implications for cross-product compatibility that a contrary ruling would have had:

> That the Lotus menu command hierarchy is a "method of operation" becomes clearer when one considers program compatibility. Under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. For example, if the user wanted the computer to print material, then the user would have to learn not just one method of operating the computer such that it prints, but many different methods. We find this absurd. The fact that there may be many different ways to operate a computer program, or even many different ways to operate a computer program using a set of hierarchically arranged command terms, does not make the actual method of operation chosen copyrightable; it still functions as a method for operating the computer and as such is uncopyrightable.[317]

In sum, the First Circuit's decision represents a very significant decision for a number of reasons. *First*, its logic suggests that computer program menu command structures may not be copyrightable in any instance. *Second*, it recognizes that one must ask a threshold question – before proceeding with an abstractions analysis or an analysis of similarity – of whether the

---

[316] Id. at 819 (citation to Autoskill omitted).

[317] Id. at 817-18.

allegedly copied elements of a computer program even constitute copyrightable *subject matter*. If they do not, no further infringement analysis need be performed.

Third, the First Circuit's decision rejects an approach taken by many courts in the computer program copyright cases (more often at the district court level) that judges whether an allegedly copied element is copyrightable principally by looking to whether there were other ways such element could have been designed or implemented, rather than looking at the inherent characteristics or nature of the element itself. The First Circuit's decision holds that if, because of its nature or characteristics, a computer program element is not within the subject matter of copyright, then it is irrelevant whether there were other ways such element could have been designed or implemented (all such ways are uncopyrightable).

Fourth, the First Circuit's decision evidences significant sensitivity to the policy implications of a decision that would impede cross-product compatibility in functional areas such as menu commands, particularly where such commands have become the basis for independently created works such as macros. As is evident in several of the other decisions discussed in this article, the courts (especially the appellate courts) seem to be increasingly sensitive to the risks of overprotection of elements of computer programs that are directly related to a program's functionality. Thus, although many of the district court decisions in recent years have continued to expand the scope of "feel" protection, one may see a trend in the next few years to reduce "feel" protection, particularly as more cases reach the appellate courts. As noted previously, such trend has already been manifest in the "look" cases.

THE SUPREME COURT'S DECISION

The Supreme Court granted a petition for certiorari filed by Lotus. On January 16, 1996, just one week after oral argument of the case, the Supreme Court affirmed the First Circuit's decision by an equally divided court in a 4 to 4 decision.[318] Justice Stevens took no part in the consideration or decision of the case.

The equally divided Supreme Court illustrates the difficulty of applying traditional copyright doctrines to computer programs, which are fundamentally functional works. Judge Keeton's decisions in the lower court and the First Circuit's decision on appeal embody fundamentally different ways of looking at what was at issue in the case. Judge Keeton defined the "idea" underlying the command hierarchy of Lotus 1-2-3 as the functions served by those commands, and viewed the command names themselves and their structure as *separable* expression, heavily based on the fact that the designers of the command hierarchy exercised considerable *choice* in choosing the names and ordering the commands.

By contrast, the First Circuit was willing to view the command names and structure as an *inseparable* part of the method of operating Lotus 1-2-3 since the command names themselves were utilized by the user to invoke the underlying functions, and at least the first letters of those commands were necessary to run macros written by users of the program. Thus, whether the

[318] 133 L.Ed.2d 610 (1996).

command names and structure were to be viewed as separable expression, or as an inseparable part of function or operation, fundamentally divided the two courts.

As will be discussed further below in connection with the Compaq case, this fundamental disagreement over whether an element of a program that has become a standard should be viewed as separable expression by virtue of the choices exercised by the original developers of that element, or should be viewed as an uncopyrightable scenes a faire, system, or method of operation, continues to divide the courts faced with "copying for compatibility" cases. Although the Borland case afforded the Supreme Court a good opportunity to address this issue head on, the equal division of the Court means that the lower courts must continue to work through this issue for perhaps years to come, as there continues to be no Supreme Court decision addressing a software copyright issue.

With respect to the specific issue raised by the Borland case of whether computer commands and command hierarchies are copyrightable, there continues to be a division of authority. In addition to the First Circuit's decision in Borland, the Mitel[319] case (discussed below) held that command codes for programming telephone call controllers constituted an uncopyrightable system or method of operation, and were also an uncopyrightable scenes a faire and could be copied under the fair use doctrine. The Mitel court rejected the plaintiff's argument that its command codes were copyrightable merely because it had exercised choices in defining them. In addition, although less directly on point, the district court on remand in the Gates Rubber[320] decision (discussed below) held that the plaintiff's menus and sorting criteria embodied in its program were an "idea," although the court's analysis of the issue is very sparse.

By contrast, the Tenth Circuit in the Autoskill[321] case (discussed in the next section) affirmed a finding of infringement based in part on similarities in the user interface in which the student selected one of three word choices appearing on the screen by hitting the 1, 2, or 3 key. A district court in the Control Data[322] case (discussed below) issued a preliminary injunction against the developer of a compatible network operating system in a case in which the plaintiff had alleged copying, among other things, of its network operating system commands and source code parameters. In addition, the Eleventh Circuit in the Bateman[323] case (discussed below) rejected the defendant's argument that "interface commands" are per se uncopyrightable, although it stated in dicta that under particular facts, copying of such commands might not constitute infringement under any of several doctrines – originality, section 102(b) of the copyright statute, fair use, copyright estoppel, or misuse.

---

[319] Mitel, Inc. v. Iqtel, Inc., 896 F. Supp. 1050 (D. Colo. 1995).

[320] Gates Rubber Co. v. Bando Am., Inc., 9 F.3d 823 (10th Cir. 1993), reversing 798 F. Supp. 1499 (D. Colo. 1992), amended, 1992 U.S. Dist. LEXIS 13601 (D. Colo. 1992).

[321] Autoskill, Inc. v. National Educ. Support Sys., 994 F.2d 1476 (10th Cir.), cert. denied, 114 S. Ct. 307 (1993).

[322] Control Data Sys. v. Infoware, Inc., 903 F. Supp. 1316 (D. Minn. 1995).

[323] Bateman v. Mnemonics, Inc., 79 F.3d 1532 (11th Cir. 1995).

Thus, for the time being, the law is unclear whether, and the extent to which, one may copy the commands and associated parameters and formats of another computer program for compatibility.

D. THE AUTOSKILL CASE

A significant pair of "feel" cases – decided exactly five months apart – issued from the Tenth Circuit in late 1992 and early 1993. These cases are important because together they adopt in the Tenth Circuit the abstraction/filtration/comparison approach of the Second Circuit's Altai case. They present a conundrum, however, in that they reach widely varying results as to the proper scope of copyright protection for the nonliteral elements of a computer program. The first of these cases, the Autoskill case, is discussed in this Section. The second of these cases, the Gates Rubber case, is discussed in the next Section.

1. Factual Background

In Autoskill, Inc. v. National Educational Support Sys.,[324] the plaintiff owned the copyright in a program called "Autoskill" for testing, diagnosing and training reading skills. The defendant studied the Autoskill program in detail, then wrote its own computer program for reading skills called "NESS." Each of the programs was based on the identification of three reading sub-types of students. The programs administered tests to determine a student's sub-type by presenting thirteen different categories of word form types based upon different combinations of consonants and vowels, ranging from one letter to four letters.

The plaintiff alleged that the defendant's program infringed the plaintiff's copyright in the "Autoskill" program, and sought a preliminary injunction. The plaintiff made no allegation that the defendant's program copied source code, but alleged that the defendant's program copied the structure, sequence and organization and "total concept and feel" of the plaintiff's program.

The district court found a number of similarities between the programs that related more to "the important pedagogical aspects of the reading program" than to "the logic flow between the display screens."[325] Based on these findings, the district court concluded that the plaintiff had established a substantial likelihood of success on its claim of copyright infringement, and issued a preliminary injunction against the defendant. The district court based its finding of infringement essentially upon the fact that the two programs implemented the same pedagogical methods of testing, diagnosing and training reading skills, and therefore had the same "feel."[326]

---

[324] 793 F. Supp. 1557 (D.N.M. 1992).

[325] The court ignored several visual differences between the two programs. NESS used prompts and color; Autoskill did not. NESS caused the word choices to appear on the screen in phases whereas all the words appeared on the Autoskill screen at the same time. The court concluded that these differences were not important or substantial parts of the Autoskill program. Id. at 1570-71.

[326] Id. at 1567-71.

2. The Tenth Circuit's Decision

On appeal, the Tenth Circuit affirmed.[327] The court noted that the district court had adopted Professor Nimmer's three-step filtration analysis for judging substantial similarity (as part of the access-plus-substantial-similarity test for proving copying), which was very similar to the Second Circuit's abstraction/filtration/comparison analysis in the Altai case. The Tenth Circuit stated, however, that it need not decide whether this test was an appropriate one for a final adjudication:

> In this preliminary injunction appeal we need not decide which is precisely the correct method of analysis for a final copyright judgment, because we are satisfied that the trial judge used a permissible method of analysis here; moreover, Autoskill showed a likelihood of success in defining as protectable the portions of its program for which the judge granted preliminary injunctive relief, after finding there was substantial similarity between the identified portions of the NESS and Autoskill programs which the judge compared. The choice of the precise test for such analysis can await an appeal requiring that choice, perhaps preferably one from a final copyright judgment.[328]

(a) Levels of Abstraction Analysis

The court noted that a substantial similarity analysis must compare "portions of the alleged infringer's works with the portions of the complaining party's works which are determined to be legally protectable under the Copyright Act."[329] The court noted that the district court had used an abstractions analysis to separate idea from expression to determine which portions of the plaintiff's works were unprotectable ideas, and which were potentially expression that must be subjected to the filtration analysis. Noting that the district court's abstractions analysis did not "reveal precisely the abstractions analysis outlined in Altai", the Tenth Circuit nevertheless felt that "the record furnishes an ample factual basis for the trial judge's analysis on the levels of abstraction and his conclusions as to which were idea levels not entitled to protection, and which were in the expression area and possibly eligible for protection after filtration analysis."[330]

---

[327] Autoskill Inc. v. National Educational Support Sys., 994 F.2d 1476 (10th Cir.), cert. denied, 114 S. Ct. 307 (1993).

[328] Id. at 1490-91. Neither party had put into issue the particular test used by the district court to judge substantial similarity. Rather, they merely disputed its application to the facts of the case in determining what similarities which protectable by copyright and which were not. See id. at 1490 n.17.

[329] Id. at 1490.

[330] Id. at 1492-93.

In particular, the court found no error in the district court's determination that the two highest levels of abstraction of the following four-level breakdown were ideas, and the remainder were potentially expression:

> (I) recognition that at highest level of abstraction both programs share common ideas of using a computer to diagnose, remediate and teach reading skills;
>
> (II) distinctions between subtypes in diagnosis and training;
>
> (III) specific subtypes diagnosis and training of Type O, A, and S; and
>
> (IV) details of testing and training.[331]

It is interesting to note that all of these levels of abstraction were described at a functional, or training, level, as were the specific similarities noted by the district court in its decision (detailed below).

(b) Filtration Analysis

Citing the Altai and Brown Bag cases, the Tenth Circuit noted that the filtration step requires the court to examine the program's structural components at each level of abstraction to determine whether they are excludable under traditional copyright doctrines in order to define the scope of the plaintiff's copyright.[332] The Tenth Circuit found that the district court had properly applied the merger and scenes a faire doctrines in its filtration analysis.[333] The Tenth Circuit rejected the defendant's argument that the features of the Autoskill program the plaintiff sought to protect were not protectable because they were drawn largely from a scholarly study of reading known as the "Doehring study" and "were common and obvious methods of testing subjects."[334] The court cited testimony that "Autoskill did not simply use Dr. Doehring's theories in a computerized form, but instead had to make 'significant changes' to Dr. Doehring's techniques to develop effective training programs."[335]

---

[331] Id. at 1493 n.20.

[332] Id. at 1493-94.

[333] The district court filtered out the 13 categories of vowel and consonant combinations used in the Autoskill program under the merger doctrine because the court found them to be dictated by teaching reading in English. The court filtered out the "silent sentence" and "silent paragraph" components of the Autoskill program under the scenes a faire doctrine. Id. at 1494.

[334] Id.

[335] Id. at 1495. As an example of originality in the Autoskill program going beyond the Doehring study, the court cited the fact that "the Autoskill system did not simply involve touching keys 1, 2, or 3, but involved looking at the word on the screen and responding with hands on the keyboard, a system that took considerable investigation and research staff work, and also that of statisticians and programmers." Id. (emphasis added).

The court also cited testimony that the Autoskill program tested for 39 subskills, which were not the same subskills tested in the Doehring study. "[The plaintiff's expert] testified that the NESS program used the same 39 subskills, which was not necessary because there were thousands of possible choices that could have been made."[336] The court did not, however, explain why the functional choice of which subskills to test for constituted copyrightable expression.

The court further cited testimony to the effect that the following other features of Autoskill went beyond the Doehring study and were not found in other software programs that perform reading analysis skills:

- The manner in which the program used the computer to record mastery and speed of response;
- The manner of continuous reinforcement of the students, offering 50 trials to reinforce a student's response;
- The presentation of graphs and the program's immediate feedback;
- The way that Autoskill assessed the abilities of students and divided them into three distinct subtypes, and then prescribed training according to those subtypes;
- An emphasis on errors as being more important than latency;

In response to the defendant's argument that this keying "system" constituted an unprotectable method under § 102(b) of the copyright statute, the court responded by stating that "we must go beyond the literal language of the statute and apply the idea/expression distinction to resolve this issue." Id. at 1495 n.23. The court noted that "non-literal aspects of computer programs certainly can be subject to copyright protection," id., and then concluded from this the following:

> We think, for the purposes of the preliminary injunction, that the record showed that the keying procedure reflected at least a minimal degree of creativity. Further, NESS has not pointed to substantial evidence in the record that this procedure was such a common practice, or that it was dictated by efficiency considerations, so that it should have been filtered out of the analysis.

Id. The court appears to have confused the issue of originality with the issue of what constitutes copyrightable subject matter. The court should first have determined whether a "keying procedure" constitutes copyrightable subject matter in view of the proscriptions of § 102(b) of the copyright statute, and then, if so, the court should have asked whether the plaintiff's expression of that subject matter was original.

[336] Id.

- A diagnostic assessment for a profile indicating a subtype of reading difficulty; and

- The use of alternating sense and nonsense words.[337]

Based on this testimony, the Tenth Circuit concluded that "the expert testimony and the exhibits furnish a substantial basis for the trial judge's conclusion that there are protectable elements in the Autoskill program that survive the filtration process."[338]

(c) Comparison Analysis

Turning to the comparison step of the analysis, the Tenth Circuit found that the district court "had an adequate basis for observing 'many significant similarities' in the protectable aspects of the Autoskill program and the allegedly infringing NESS program"[339] based upon the following similarities in the two programs:

- Asking the student to read a word on the screen orally, and having the trainer decide whether or not the word is read correctly and record speed and accuracy into the computer;

- In the audio identification test, having three word choices appear on the screen and an auditory stimulus of a target word or nonsense word be presented to the student (the student selects the word he hears and indicates a response by hitting the 1, 2, or 3 key);

- In the visual identification test, the screen displays four words or nonsense words, and the target word is isolated from the other words and the student is expected to choose one of the remaining three words which is identical to the target word;

- Use of alternating words and nonsense words;

- Recording speed, or latency of response, and accuracy data, and use of the information in the same manner;

- Training students according to the same three testing topics;

- Providing immediate feedback to students about accuracy;

- Use of similar criteria for a student's progressing to the next subprogram;

- Presentation of skills hierarchically from the simple to the complex;

---

[337] Id. at 1495-96.

[338] Id. at 1496.

[339] Id. at 1496-97 (quoting Autoskill, 793 F. Supp. at 1569).

- Administration of a visual scanning test in combination with the other tests to determine the student's subtype;

- Recording the student's progress within each section in matrices; and

- Using graphs for the same purpose.[340]

The Tenth Circuit endorsed the district court's rejection of the defendant's expert witness' testimony concerning differences in "logic flow between the display screens of both programs" and in visual differences in the screens themselves, because such differences were "not pedagogically significant."[341] In sum, both the district court and the Tenth Circuit appear to have based their infringement analysis almost entirely on similarities in the pedagogical aspects, and combinations of methods, of the two programs. It appears that the practical effect of the final decision may have been to protect the training methods themselves, or at least the particular combination of methods adopted by the plaintiff in its program.

Thus, although the Tenth Circuit laid the groundwork in the Autoskill case for adoption of an abstraction/filtration/comparison test for copyright infringement, its application of that test afforded a very broad scope of protection to the plaintiff's program and its embodied techniques.

E. THE GATES RUBBER CASE

The second recent "feel" case to come out of the Tenth Circuit, Gates Rubber Co. v. Bando American, Inc.,[342] formally adopted the analytical filtration approach of the Altai case. The Tenth Circuit had previously stated in the Autoskill case that it was not deciding what test should be applied for "analysis for a final copyright judgment," but rather was adopting an approach similar to that of Altai for purposes of resolving a motion for a preliminary injunction.[343] In Gates Rubber, the Tenth Circuit made the filtration approach of Altai the "final" test for copyright infringement.

The importance of the Tenth Circuit's decision in Gates Rubber is enhanced by the fact that it reversed a lower court's ruling that explicitly disagreed with the approach adopted by the Altai case, and that would have expanded considerably the scope of protection that may be afforded to nonliteral computer program elements – in particular, the "dynamic" behavior of a computer program as it operates. In addition, the Tenth Circuit's application of the test in Gates Rubber afforded a much narrower scope of protection to the plaintiff's computer program than did the application of a similar test in Autoskill. Accordingly, one will have to wait for additional cases out of the Tenth Circuit to see how broadly or narrowly the newly adopted infringement test will be applied.

---

[340] Id. at 1497.

[341] Id.

[342] 9 F.3d 823 (10th Cir. 1993), reversing Gates Rubber Co. v. Bando American, Inc., 798 F. Supp. 1499 (D. Colo. 1992), amended, 1992 U.S. Dist. LEXIS 13601 (D. Colo. 1992).

[343] 994 F.2d 1476, 1491 (10th Cir. 1993).

1. Background on the Lower Court's Decision

The plaintiff Gates Rubber Co. was the owner of a computer program known as Design Flex 4.0 designed to aid in the selection of replacement industrial belts. The defendant company was formed by a former employee of Gates, who in turned hired away a couple of other employees from Gates. The defendant produced its own industrial belt computer program, which the plaintiff alleged infringed its copyright in Design Flex. Although there was no evidence that the defendant had engaged in any literal copying of source code or object code, a central issue in the case concerned some calculation methods based upon certain mathematical constants used by Design Flex to design a belt drive and determine belt size.

The district court began its analysis with a review of several infringement tests developed by courts in various other Circuits, all within the framework of a basic two-step test in which expert testimony is relied upon in step one to analyze similarities objectively, followed by a second step in which overall substantial similarity is judged without the aid of expert testimony. The court concluded that these traditional tests were unsatisfactory as applied to computer programs, in view of the technical and legal complexities involved, and that substantially greater emphasis should be afforded to expert testimony at all stages of the analysis.[344]

Accordingly, the court ruled that it should first hear expert testimony as to all similarities between the plaintiff's and defendant's programs, whether protectable or not, and then apply the abstractions test (and other "limiting" doctrines such as merger and scenes a faire) at the substantial similarity phase "once the factual considerations by the experts are taken into account, to consider as a matter of law, which portions of the [plaintiff's] program are properly protectable."[345]

Applying its test, and based upon the testimony of three experts in the case, the court compiled the list of similarities set forth in the table below, and made the rulings indicated beside each entry as to whether or not it constituted protectable expression:

TABLE III – ELEMENTS ADJUDICATED FOR COPYRIGHTABILITY

FEATURE	LEVEL OF SIMILARITY	PROTECTABLE?

[344] 798 F. Supp. 1499, 1511, 1513-14 (D. Colo. 1992).

[345] Id. at 1514. The court further noted that it was "of the opinion that it is far preferable, especially in an area of legal and technological sophistication as complex as this area of copyright protection, to draw upon a larger arsenal of facts in order to design or derive the appropriate legally significant facts. Once these are gathered and expert testimony is heard, the court can then analyze which portions of the program, according to the expert testimony, infringes the protected expression." Id. at 1511.

Object and source code	No evidence of copying; no similarity	Yes, but not similar
Menus	Substantially similar	Unclear
Formulas	Substantially similar	No (previously published)
Mathematical constants	Identical	Yes
Data flow	Substantially similar	Yes
Control flow (sequence of events)	Substantially similar	Yes
Programming style	Differences, consistent with independent development	Yes, but not similar
Level of complexity	Expressions used have similar "look and feel"	No
Install files	Substantially similar, nearly verbatim copy	Yes
Engineering calculation modules	Substantially similar behavior	Yes
Design modules (V-belt algorithm)	Similar overall structure and organization	Yes
Fundamental tasks	Substantially similar	Yes
Sorting criteria (organization of data)	Substantially similar	Yes
Common errors/misbehaviors	Substantially similar	Yes

From the table above, it is apparent that the district court extended copyright protection to a great deal of the "dynamic" behavior and operational control flow of the plaintiff's program, including mathematical algorithms, mathematical constants, and choice of functions the program was to perform, as well as to the more traditional modular "static" structure of the code itself. Indeed, the court explicitly concluded that "behavior" of a program can be protectable "expression":

> The Court will respectfully disagree with the Altai decision and hold that a program's behavior can be protected by copyright law. A particular example of common error concerns the minimum/maximum error where both programs, upon receiving a particular answer, erroneously take the user back to another part of the program. In this example, the commonality of this error denotes "behavior" as to how one part of the program works with another. This is part of the creative expression of the program itself.[346]

At the time of its decision, the district court's ruling in Gates Rubber represented one of the broadest scopes of protection afforded to a computer program under copyright law, concluding that because the two programs at issue "felt" and operated or "behaved" the same,[347] there was sufficient basis for a finding of copyright infringement.

2. The Tenth Circuit's Decision

On appeal, the Tenth Circuit vacated the district court's decision and remanded for further proceedings. The court concluded that the district court erroneously extended copyright protection to certain unprotectable elements of a computer program and that it failed properly to determine the protectability of many of the elements of the plaintiff's program which it found had been copied by the defendants.

The Tenth Circuit's opinion contains an exhaustive analysis and synthesis of the various recent Circuit court opinions dealing with the scope of protection for computer programs and the various tests for adjudicating infringement.[348] The Tenth Circuit sets forth a clearly delineated test for copyright infringement of a computer program, at least in a case primarily directed toward infringement of code (as opposed to user interface).

The Tenth Circuit's test is formulated upon the following two separate inquiries:[349]

1) "Factual Copying": Whether the defendant, as a factual matter, copied portions of the plaintiff's program. The court referred to this issue as one of

---

[346] Id. at 1518-19.

[347] "[T]he parties did undertake a demonstration of the two programs, and the undersigned judge, being largely unfamiliar with computers and their processes, will make only this brief observation. The Court did note significant similarities in the running of the two programs, while the appearance of the screens was different, the content and method of proceeding through calculations were quite similar, as was the overall operation of the two programs." Id. at 1516.

[348] "We find in these and other cases that have considered the copyrightability of computer programs that there has begun to be developed a coherent approach to the protectability analysis. The approach that we outline today is consistent with this evolving approach to the copyright protection of computer programs." 9 F.3d at 841-42.

[349] Id. at 832.

determining whether "factual copying" took place.[350]

2) Unlawful Appropriation: Whether, as a mixed issue of fact and law, those elements of the program that have been copied are protected expression and of such importance to the copied work that the appropriation is actionable.

Each of these inquiries is treated separately below.

(a) Factual Copying

The court noted that the plaintiff can establish factual copying either by direct evidence, or (more commonly) by indirect evidence by showing that the defendant had access to the copyrighted program and that there are probative similarities between the copyrighted material and the allegedly copied material.[351] "Ultimately, to prove factual copying, the plaintiff must come forward with sufficient evidence that a reasonable factfinder, taking together the evidence of access and the similarities between the programs, could find that the second work was copied from the first."[352] Establishment of a prima facie case of copying through the indirect method, however, merely creates an inference that the defendant copied from the plaintiff's program. The defendant can come forward with evidence of independent creation to rebut the inference.[353]

With respect to the issue of establishing factual copying, the Tenth Circuit made the following two observations:

- *First*, the court noted that the issue of factual copying need not necessarily be determined in every case before the issue of whether the copied elements are protectable: "Although we suggest that it will often be helpful to make an initial determination of whether the defendant copied portions of the plaintiff's program before determining whether the copying involved protectable elements under the copyright law, there may be cases where the issue of protectability can more efficiently be addressed first. The order of the analysis will depend on the individual facts and issues in each case."[354]

---

[350] Id. at 833.

[351] Id. at 832.

[352] Id. at 833. "The degree of similarity between programs necessary to give rise to the inference that copying occurred will vary from case to case. A high degree of similarity may permit access to be inferred. Conversely, where there is strong proof of access, the necessary showing of factual similarity will be relatively lower. However, we note that no matter how conclusive proof of access may be, liability may not attach without some showing of similarity." Id. at 833 n.9 (citations omitted).

[353] Id. at 833 n.8.

[354] Id. at 833.

- *Second*, the court noted that, where the issue of factual copying is determined first, it will often be helpful to consider similarity of both protectable and unprotectable elements as probative of the issue: "We acknowledge that unprotectable elements of a program, even if copied verbatim, cannot serve as the basis for ultimate liability for copyright infringement. However, the copying of even unprotected elements can have a probative value in determining whether the defendant copied the plaintiff's work. Where a court first extracts all unprotected elements of a work, and only compares protected elements, it deprives itself of the use of probative, and potentially essential, information on the factual issue of copying. ... The fact that non-protectable elements of the original program were also copied, although it cannot be the basis for liability, can be probative of whether protected elements were copied."[355]

(b) Unlawful Appropriation

If factual copying by the defendant is established, in order to impose liability for unlawful appropriation, the court must find "that the defendant copied protectable elements of the plaintiff's program and that those protectable elements comprise a substantial part of the plaintiff's program when it is considered as a whole."[356] The Tenth Circuit adopted the three-part test of the Computer Associates v. Altai case (which the court noted had already been previously approved for use five months earlier in the Tenth Circuit's decision in the Autoskill case) for separating protectable elements from unprotectable elements that may have been copied:

1) Step One: Abstraction. The court must first "dissect the program according to its varying levels of generality as provided in the abstractions test."[357]

2) Step Two: Filtration. "[P]oised with this framework, the court should examine each level of abstraction in order to filter out those elements of the program which are unprotectable."[358]

3) Step Three: Comparison. "[T]he court should then compare the remaining protectable elements with the allegedly infringing program to determine whether the defendants have misappropriated substantial elements of the plaintiff's program."[359]

---

[355] Id. at 832 n.7. Elsewhere the court noted that "an initial holistic comparison may reveal a pattern of copying that is not obvious when only certain components are examined." Id. at 841.

[356] Id. at 833.

[357] Id. at 834.

[358] Id.

[359] Id.

Each of these steps is elaborated below.

(c) Abstraction

The first step of the Abstraction/Filtration/Comparison test involves dissecting the program into its various levels of abstraction "in a way that parallels the typical development of a program."[360] The court noted that given the ever-changing nature of computer technology, no strict methodology for abstraction could be set forth, but that a computer program can often be parsed into at least six levels of generally declining abstraction:

(i) the main purpose

(ii) the program structure or architecture

(iii) modules

(iv) algorithms and data structures

(v) source code, and

(vi) object code.[361]

The court noted that expert testimony will ordinarily be helpful to organize a particular program into various levels of abstraction.[362]

(d) Filtration

Once the various levels of abstraction have been identified, the court must filter out those elements of the program that are not protected by copyright, based upon the following copyright doctrines:

1) The Idea-Expression Dichotomy. The court must use the various levels of abstraction into which the program has been dissected to decide what levels constitute unprotectable ideas. "[T]he main purpose or function of a program will always be an unprotectable idea. Likewise, each module may typically be described by its individual purpose or function, and the basic function or purpose of a module will nearly always be an unprotectable idea or process. At the other end of the abstractions spectrum, source and object code, which are the literal elements of a program, will almost always be found to be protectable expression unless the doctrines of merger and scenes a faire come into play. The intermediate levels of abstraction, such as structure, sequence,

---

[360] Id. at 834.

[361] Id. at 834-35.

[362] Id. at 835.

organization, and the like, are less prone to generalizations."[363]

2) The Process-Expression Dichotomy. Copyright cannot extend to processes embodied in the program. "Most commonly, processes will be found as part of the system architecture, as operations within modules, or as algorithms."[364]

3) Facts. "In computer programs facts may be found at a number of levels of abstraction, but, will most often be found as part of data structures or literally expressed in the source or object codes."[365]

4) Public Domain. "[A] court must filter out all unoriginal elements of a program, including those elements that are found in the public domain."[366]

5) The Merger Doctrine. The court noted that copyright protection must be denied to "expression that is inseparable from or merged with the ideas, processes, or discoveries underlying the expression," but did not state how this doctrine might apply specifically to computer programs.[367]

6) Scenes a Faire. The court noted that the scenes a faire doctrine excludes from protection expressions standard or common to a particular topic, as well as "those elements of a program that have been dictated by external factors":

→ "hardware standards and mechanical specifications"

→ "software standards and compatibility requirements"

→ "computer manufacturer design standards"

→ "target industry practices and demands"

→ "computer industry programming practices"[368]

---

[363] Id. at 836. The court noted the heavy criticism the Whelan case had received for defining the idea of a computer program as its purpose or function, and agreed that such criticism is valid if the opinion is read to imply that a computer program can have only one idea. Id. at 840 n.17. The court further noted, however, that so long as one understands that a computer program may have more than one idea, Whelan's "conclusion that the structure of a program may be protectable is sound." Id. at 840.

[364] Id. at 837.

[365] Id.

[366] Id.

[367] Id. at 838.

[368] Id.

> Interestingly, the court also noted that "the *scenes a faire* doctrine may implicate the protectability of interfacing and that this topic is very sensitive and has the potential to effect widely the law of computer copyright. This appeal does not require us to determine the scope of the *scenes a faire* doctrine as it relates to interfacing and accordingly we refrain from discussing the issue."[369]

(e) Comparison

After the court has filtered out the unprotectable elements of the program, "it is left with a core of protected elements that can be compared to the alleged infringing program. Ultimately the court must decide whether those protectable portions of the original work that have been copied constitute a substantial part of the original work – i.e. matter that is significant in the plaintiff's program."[370]

An important point to note about this third step of the Tenth Circuit's Abstraction/Filtration/Comparison test is that the court does not state specifically how the "core of protected elements" is to be compared to the allegedly infringing program. Specifically, is a substantial similarity test used a second time in making the comparison, as was done in the Borland decisions discussed above? Or, having determined that copying took place through a substantial similarity test in the factual copying inquiry, is the only inquiry to be made in the final comparison step whether what was copied constitutes "a substantial part of the original work"?

Unfortunately, the Tenth Circuit was not called upon in the Gates Rubber case to explicate the details of this issue, for the court found that the district court had failed to undertake a proper filtration analysis with respect to several elements and had erroneously found other elements to be protectable in the filtration step. Accordingly, the court never reached the comparison step of the test. The Second Circuit's description of the comparison step in the Altai case suggests a substantial similarity test is indeed to be applied.[371] Because the Tenth Circuit

---

[369] Id. at 838 n.14.

[370] Id. at 838-39.

[371] The Second Circuit described its comparison step thusly: "Left with a kernel, or possible kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program. The result of this comparison will determine whether the protectable elements of the programs at issue are substantially similar so as to warrant a finding of infringement." Altai, 982 F.2d at 706 (emphasis added).

What is, unfortunately, unclear from the Second Circuit's opinion in Altai is whether substantial similarity is also used prior to the Abstraction/Filtration/Comparison analysis to establish factual copying. The Altai opinion does recognize that the plaintiff must prove "copying," and that copying may be established by indirect evidence through proof of access plus substantial similarity. Id. at 701. It is not clear whether, by this reference to "copying," the Second Circuit meant factual copying, as the Tenth Circuit uses the term, or illicit

largely adopted the Altai test, one may infer that the Tenth Circuit would also apply a substantial similarity test of some sort in the comparison step, but this issue is, unfortunately, not explicitly fleshed out in the court's opinion in Gates Rubber.

3. Application of the Test

The Tenth Circuit reviewed the district court's decision against its detailed formulation of its test for infringement. The Tenth Circuit found that the district court had properly conducted an initial inquiry into whether there had been factual copying, using an access-plus-substantial-similarity test. The Tenth Circuit found that the district court had next properly attempted to apply the abstractions test to identify unprotectable ideas through dissection. The Tenth Circuit noted that, in applying the filtration step, the district court had concluded that the merger doctrine and the scenes a fair doctrine were inapplicable, and that the district court had then evaluated what it regarded as the protectable elements and determined that they were sufficiently significant to the plaintiff's program for a finding of unlawful appropriation.[372]

The Tenth Circuit concluded, however, that the district court "failed to undertake a proper filtration analysis with respect to several elements and that it erroneously found other elements to be protectable":[373]

---

appropriation (a legal conclusion). The confusion stems from the fact that the Second Circuit described its Abstraction/Filtration/Comparison test as a test to "ascertain[] substantial similarity," which implies a legal conclusion, not a factual inquiry. Id. at 706. One might infer from this description of the test that substantial similarity is to be used only to form the legal conclusion of illicit copying, and not in the inquiry with regard to factual copying.

Later in the opinion, however, in discussing the role of expert testimony, the Second Circuit implicitly recognized that an inquiry with regard to factual copying is to be made: "As a threshold matter, expert testimony may be used to assist the fact finder in ascertaining whether the defendant had copied any part of the plaintiff's work." Id. at 713. This is clearly a reference to factual copying, for the Second Circuit goes on to state, "once some amount of copying has been established, it remains solely for the trier of fact to determine whether the copying was 'illicit.'" Id. However, the Second Circuit does not state whether a test of substantial similarity is to be used to decide whether such factual copying took place. Rather, the court states only the following: "To this end, 'the two works are to be compared in their entirety ... [and] in making such comparison resort may properly be made to expert analysis...." Id. (quoting 3 M. Nimmer, *Nimmer on Copyright* § 13.03[E][2], at 13-62.16).

In sum, to distill out the complexity of the issue, one might simply say that if the Second Circuit's Altai opinion is read to invoke substantial similarity twice (for both the factual copying and the illicit copying inquiries), then one may infer that the Tenth Circuit would also probably invoke substantial similarity twice.

[372] 9 F.3d at 842.

[373] Id. The Tenth Circuit endorsed, however, the district court's rulings that the formulas and the level of complexity of the plaintiff's program were not protectable. Id. at 842 n.19.

(a) Constants. The Tenth Circuit held that the constants used in the plaintiff's program were unprotectable by copyright because they "represent scientific observations of physical relationships concerning the load that a particular belt can carry around certain sized gears at certain speeds given a number of other variables."[374]

(b) Menus and Sorting Criteria. The Tenth Circuit held that it could not determine whether the "menus" and the "sorting criteria" of the plaintiff's program were protectable because the district court did not define what it meant by those terms.[375]

(c) Control and Data Flow.[376] The Tenth Circuit expressed concern about the "district court's failure to examine the control and data flow in light of the process-expression dichotomy," and remanded for reconsideration on that basis.[377]

(d) Engineering Calculation and Design Modules. The district court had held that these modules constituted expression rather than idea because of "particular elements which perform in similar manners."[378] The Tenth Circuit noted that the district court failed to identify what those "particular elements" were, and remanded for further consideration to determine whether those elements were ideas, processes, or facts, or were standard in the industry and therefore scenes a faire.[379]

(e) Common Errors. The plaintiff's and defendant's programs both contained errors in the form of attempting to compute a maximum center distance that is less than a minimum center distance, and jumping to the wrong menu if the cursor happened to be over a certain character during one of the input sequences. The Tenth Circuit noted that, although the presence of common errors may be probative on the issue of factual copying, "[e]rrors per se are not protectable, although the expression containing the error may be protectable if it otherwise meets the test for protectability set forth in this opinion." Accordingly, the Tenth Circuit remanded for the district court to perform the necessary analysis of whether there was protectable expression involved in the common errors.[380]

(f) Fundamental Tasks. The Tenth Circuit noted that the term "fundamental tasks" would ordinarily refer to the highest level of abstraction of a program – the ideas or purposes underlying it – which are unprotectable. The district court stated, however, that it was using the term to be "more specific due to the types of tasks which were available to achieve the particular

---

[374] Id. at 842-43.

[375] Id. at 843.

[376] The Tenth Circuit defined "control flow" as "the overall sequence of actions and events in a program." It defined "data flow" as "the sequence of actions taken on each piece of information, that is, how the data travels through the program." Id. at 844.

[377] Id.

[378] Gates Rubber, 798 F. Supp. at 1518.

[379] 9 F.3d at 845.

[380] Id.

end of designing belts and drives."[381] The Tenth Circuit noted that it could not understand precisely what the district court meant by the term, and the district court had not elaborated upon its conclusion that the merger doctrine was inapplicable. Accordingly, the Tenth Circuit remanded for further analysis of this element by the district court.[382]

(g) Install Files. The install files constituted separate utility programs that were used to load the program from a floppy disk onto a hard disk. The Tenth Circuit noted that because the install files were part of the Disk Operating System and not a part of the plaintiff's program, it was unclear whether the plaintiff had a copyright claim on them. Accordingly, the Tenth Circuit remanded "for a determination of whether Gates held a copyright on the install files and for a reconsideration of the install files in light of the test we have set forth herein."[383]

3. The District Court's Decision on Remand

On remand,[384] the district court noted that it had relied heavily in its original finding of infringement on the copying of the constants used in the plaintiff's program. The court noted that, in view of the fact that the Tenth Circuit had held the constants not protectable, it would be unlikely that the plaintiff could otherwise establish sufficient similarity of protectable expression for a finding of infringement. Nevertheless, the court proceeded to apply the filtration step to each of the individual program elements with respect to which the Tenth Circuit had remanded for clarification:

(a) Menus and Sorting Criteria. The court clarified that by "menus and sorting criteria" it was referring to the code that created the visual displays, not the visual displays themselves (the plaintiff had waived any claim regarding copyright infringement of the screen displays). The court, citing the First Circuit's decision in Lotus v. Borland for support, ruled that, because the number of choices available for sorting criteria was small, the menus and sorting criteria should be considered to be closer to the "idea" end of the abstraction spectrum and therefore not protectable.

(b) Control and Data Flow. The court determined that the control and data flow aspects of the program's structure should be considered to "fall close to the process end of the process-expression dichotomy," and that the control and data flow also reflected standard techniques.[385] Accordingly, the court ruled that the control and data flow of the plaintiff's program was not protectable.

---

[381] Gates Rubber, 798 F. Supp. at 1519.

[382] 9 F.3d at 846.

[383] Id.

[384] Gates Rubber Co. v. Bando Chemical Industries, Ltd., Civ. Action No. 92-S-136 (D. Colo. June 12, 1995).

[385] Id. at 8.

(c) Engineering and Calculation Design Modules. The court concluded that a program to design industrial belts must of necessity perform the input-output calculations of the engineering calculation and design modules. Accordingly, the court concluded that similarities in these modules were not protectable under the process-expression dichotomy and the merger doctrine.[386]

(d) Common Errors. The court clarified on remand that its reference to "common errors" was really meant to refer to common misbehaviors of the plaintiff's and defendant's programs. With respect to such, the court ruled that the plaintiff's statement that the common misbehaviors were not qualitatively important to its program must be treated as a waiver of its argument of infringement based on the same. In addition, the court ruled that such misbehaviors were not protectable expression because there was no evidence that they were caused by similar errors in the source code.[387]

(e) Fundamental Tasks. The court held that its reference to "fundmental tasks" in its earlier opinion had been a reference to the high level functions served by the plaintiff's program and that, because such tasks exist at the highest level of abstraction, they should have been filtered out of the analysis as unprotectable ideas.[388]

(f) Install Files. On remand, the court noted that the plaintiff had abandoned any claim to infringement as to the install files. In addition, the court concluded that the install files were not part of the plaintiff's program and that, even if they were, they "are a necessary part of it and are therefore excludable under the process-expression dichotomy (or at least the merger doctrine)."[389]

In conclusion, the court held that, because all alleged similarities had been ruled unprotectable in the filtration step, there was nothing left to compare in the comparison step, and the plaintiff had therefore not established copyright infringement.[390]

F. THE CONSUL TEC CASE

Several other "feel" cases decided in the last few years have extended broad copyright protection to nonliteral elements of structure, file formats, input formats, and the user interface of computer programs. The first of these, Consul Tec Inc. v. Interface Sys.[391] involved a computer program called "3780Plus," which was a communications program enabling the transmission of information between computers. The defendant marketed a program called "3780Fast" as a "3780Plus clone." The defendant's program was designed to run the same commands as the

---

386 Id. at 9.

387 Id. at 10.

388 Id. at 11.

389 Id. at 12.

390 Id. at 12-13.

391 22 U.S.P.Q.2d (BNA) 1538 (E.D. Mich. 1991).

plaintiff's program, and to be able to execute the job files and configuration files of the plaintiff's program.

The court defined the idea underlying the plaintiff's program at a very high level of abstraction as "to enable two computers to 'communicate' with one another and to permit the user to transmit information from one computer to another."[392] The court concluded that the plaintiff's expression of this idea included "its unique compilation of commands, its command line syntax, and its status message codes",[393] and that these elements had been copied by the defendant. The court also found "[s]triking similarity ... between the invocation lines, job files, and log files of the two programs."[394]

The court rejected the defendant's defense based on "commercial necessity." The defendant argued that it had made 3780Fast look and feel like 3780Plus because its customers were accustomed to 3780Plus and would accept no alternatives. The court, citing the Paperback case, rejected this argument. Thus, the court refused to treat the expectations or training of an installed based of users as an "externality" that would limit copyright protection. The court also rejected the defendant's argument that all it had done was to create a program that merely "translates" 3780Plus commands. The court held that, because the defendant had created a program that would "behave in the same manner as 3780Plus ... [t]his 'translation' feature does not prevent a finding of substantial similarity".[395]

In a significant additional holding, the court ruled that "the 3780Fast user manual infringes [the plaintiff's] copyright in the 3780Plus user manual, because it contains references to and explanations of the commands, status codes and job files that have been found to infringe [the plaintiff's] copyright in 3780Plus."[396] Thus, if a defendant's user manual explains various nonliteral elements of a plaintiff's program that the defendant's program is designed to be compatible with, and those nonliteral elements are found copyrightable, then the defendant's user manual may also be infringing.

G. THE CMAX CASE

In the case of CMAX/Cleveland, Inc. v. UCR, Inc.,[397] the plaintiff was the owner of a computer program called "RMAX," which was designed to enable users to input, store, process and retrieve information incident to the "rent-to-own" furniture and appliance business, including inventory, rental agreement and accounting information. The defendant, originally a licensee of RMAX, set out to develop an in-house program that would perform the same functions as RMAX. In order to remain compatible with its existing RMAX data files, the defendant studied

---

[392] Id. at 1541.

[393] Id.

[394] Id.

[395] Id. at 1542.

[396] Id.

[397] 804 F. Supp. 337 (M.D. Ga. 1992).

the file structure and file names of RMAX in detail and replicated them in its own program. The defendant also copied many of RMAX's screens and reports.

The plaintiff sued for copyright infringement, arguing that under the test of Whelan, the defendant had copied copyrightable nonliteral elements of its program. The court rejected the Whelan test as overly simplistic, and adopted instead the three-step test of Altai. Applying that test, the court ruled that the file layouts (including the field definitions contained therein), record layouts, file names, naming conventions, transaction codes,[398] screens and reports of the plaintiff's program constituted protectable expression because they were neither dictated by industry standards, by efficiency, nor by the need for the program to interact with the central host computer with which it was designed to operate. The court also refused to recognize the defendant's employees' training in the use of the RMAX transaction codes as an externality that would limit protection of the codes.[399]

H. THE COMPREHENSIVE TECHNOLOGIES CASE

In contrast to the preceding two cases, the court in Comprehensive Technologies Int'l v. Software Artisans Inc.[400] refused to extend protection to certain nonliteral elements of a computer program, apparently based on certain undefined externalities of the industry. The court rejected the Whelan test and refused to find infringement where the reason "that there are similarities is that computer programs of the type involved here by necessity must use many normal, typical, generic terms and methods."[401] With very little analysis, the court simply ruled that "the use of similar terms for similar functions, the call definitions, the check boxes, the implementation and use of the same system for tracking forms, and the lack of design documentation do not persuade the court that there was copying. Again these are similarities, but are elements and use common to all computer programs of this type, as is their arrangement and the way in which they interact."[402]

I. THE ENGINEERING DYNAMICS CASE

The case decided during the last few years that affords perhaps the broadest protection to nonliteral elements of a computer program is the Fifth Circuit's decision in Engineering Dynamics, Inc. v. Structural Software, Inc.[403] In an ironic twist, this case allowed Engineering Dynamics, Inc. (EDI) to assert copyright protection over a set of input formats derived from input formats that it had successfully argued fifteen years earlier were not copyrightable in a case

---

398 "A transaction code is a randomly selected, alphanumeric sequence of characters that indicates to the computer what steps should be executed in a given situation or 'transaction' when the code is transmitted." Id. at 349 n.8.

399 See id. at 355.

400 Civil No. 90-1143-A (E.D. Va. June 2, 1992).

401 Id. at 6.

402 Id. at 8.

403 26 F.3d 1335 (5th Cir. 1994).

in which it was sued for copying of such formats, Synercom Technology, Inc. v. University Computing Co.[404] The Fifth Circuit reversed the district court's holding that computer input and output formats are not copyrightable, and remanded for a determination of whether there was infringement.

1. Background on the Litigation

In 1970, Synercom Technology, Inc. (Synercom) brought to market a computer program called STRAN, designed to solve engineering structural analysis problems. The program defined a specific set of input formats and input sequences in which the user had to input a large amount of data, including construction details and anticipated environmental and other external forces that would act upon a structure to be analyzed. In 1975, EDI entered the market with its competing program, SACS II, which utilized precisely the same input formats and input sequences as Synercom's STRAN program. EDI stressed the complete compatibility of SACS II with input data for the STRAN program, which was entered into the computer via decks of 80-column keypunch cards. Synercom sued EDI alleging copyright infringement of its input formats and sequences, but then-district judge Higginbotham ruled that such formats and sequences were not copyrightable.[405]

Over the years, EDI revised and enhanced its input formats and changed the name of its program to SACS IV. Although keypunch cards eventually ceased being used, EDI retained the 80-column data input format for its program, requiring that users store input data in such format as image files on a floppy disk. The SACS IV input formats instruct the user to place specific kinds of information in specific places within the 80-column "card" (image file). The first five columns or so are reserved for identification of the card by its name, *e.g.*, WAVE.[406] Figure 3 shows EDI's WAVE card, and Figure 4 shows the WAVE card of the defendant, Structural Software, Inc. (SSI).

---

[404] 462 F. Supp. 1003 (N.D. Tex. 1978).

[405] See Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D. Tex. 1978).

[406] The WAVE card is used to calculate ocean wave forces on structures built offshore. The placement of the required information on the proper card and in the proper columns is crucial to obtaining correct results.

EDI WAVE CARD

EDI SAMPLE PLATFORM MODEL -- ELASTIC ANALYSIS PAGE 35

SACS-II SYSTEM MEMBER DETAIL REPORT

MEMBER	GRP	LOAD CASE	DIST FROM END (FT)	FORCE FX KIPS	MOMENT MY IN-KIPS	MOMENT MZ IN-KIPS	SHEAR FY KIPS	SHEAR FZ KIPS	TORSION MX IN-KIPS	AXIAL STRESS KSI	BENDING STRESS Y KSI	BENDING STRESS Z KSI	COMB. STRESS KSI	SHEAR STRESS KSI	MAX COMB. UNITY CHECK
101- 201	LG1	5	0.0	11.17	469.8	-95.7	.08	.93	-5.4	.16	.61	-.86	.77	.02	.026
		6		11.53	470.1	-250.7	-10.05	.95	488.1	.16	.62	-.34	.67	.45	.028
		7		243.88	129.3	-230.0	-7.15	-6.41	136.8	3.48	.17	-.38	3.82	.23	.132
		5	34.0	18.18	597.6	-14.1	.08	-.38	-5.4	.26	.78	-.02	1.09	.01	.045
		6		18.46	682.3	-1749.4	3.38	-.35	488.1	.26	.78	-2.37	2.67	.36	.085
		7		249.93	-895.5	-1281.6	2.41	1.76	136.8	3.58	-1.16	-1.67	5.61	.13	.188
		5	68.0	25.02	225.8	17.6	.08	-1.52	-5.4	.36	.29	.02	.65	.03	.029
		6		25.38	168.4	2881.8	19.96	-1.84	488.1	.36	.21	3.73	4.12	.40	.131
		7		256.85	1075.3	2053.7	14.32	12.19	136.8	3.67	2.44	2.67	7.29	.36	.242
201- 301	LG1	5	0.0	21.82	-1232.5	11.2	-.03	3.98	-8.8	.31	-1.63	.01	1.96	.06	.063
		6		26.43	-1235.5	2888.7	-22.66	4.18	-603.6	.38	-1.61	3.64	6.36	.72	.133
		7		660.18	584.3	2288.4	-16.91	-9.96	-429.1	9.44	.66	2.87	12.39	.56	.421
		5	27.9	27.51	-85.6	.9	-.03	2.98	-8.8	.39	-.11	.00	.50	.05	.025
		6		32.12	-82.9	-2033.8	-5.48	2.76	-603.6	.46	-.11	-2.59	3.11	.48	.103
		7		665.86	-1075.9	-1498.8	-4.65	1.83	-429.1	9.53	.87	-1.94	11.42	.35	.485
		5	55.9	33.19	744.6	-9.4	-.03	1.97	-8.8	.47	-1.98	-.01	1.44	.03	.063
		6		37.88	682.3	-392.8	15.99	1.31	-603.6	.56	.78	-.51	1.46	.62	.043
		7		671.54	1329.3	-564.4	10.68	15.82	-429.1	9.61	1.99	-.73	11.73	.56	.481
301- 401	LG3	5	0.0	27.67	-278.5	-17.5	-.04	-.52	-29.8	.40	-.36	-.02	.76	.03	.015
		6		30.94	-386.9	-585.6	-8.44	.77	-1341.6	.56	-.50	-.76	1.47	.99	.052
		7		971.98	680.9	-551.4	-6.99	-14.27	-589.0	13.91	.78	-.72	14.97	.60	.515
		5	11.8	38.22	-382.9	-22.6	-.04	-.97	-29.8	.43	-.50	-.03	.93	.03	.041
		6		41.50	-323.0	-1038.7	2.91	.89	-1341.6	.59	-.62	-1.35	2.01	.91	.065
		7		976.54	-936.2	-727.4	3.12	-6.78	-589.0	13.94	-1.22	-.05	15.48	.49	.533
		5	23.7	32.58	-547.0	-27.8	-.04	-1.39	-29.8	.47	-.71	-.04	1.16	.04	.052
		6		43.86	-351.7	116.2	10.51	-.55	-1341.6	.63	-.46	.15	1.11	1.08	.055
		7		976.90	-1108.4	245.0	11.48	.92	-589.0	13.98	-1.03	.32	15.81	.55	.544
		5	23.7	32.58	-547.0	-27.8	-.04	-1.39	-29.8	.31	-.47	-.02	.79	.03	.030
		6		43.86	-351.7	116.2	10.51	-.55	-1341.6	.42	-.30	.10	.74	.72	.017
		7		976.90	-1108.4	245.0	11.48	.92	-589.0	9.26	-1.21	-.21	10.58	.36	.361
		5	36.7	36.89	-768.3	-32.6	-.04	-2.01	-29.8	.36	-.66	-.03	1.03	.05	.046
		6		47.37	-476.2	2798.6	27.05	-1.41	-1341.6	.45	-.41	2.48	2.89	.84	.093
		7		980.48	-798.6	2292.6	28.65	9.39	-589.0	9.30	-.68	1.98	11.35	.47	.393
		5	45.7	39.53	-1069.2	-37.4	-.04	-2.62	-29.8	.37	-.97	.05	1.38	.04	.055
		6		50.81	-711.9	7198.2	48.48	-2.24	-1341.6	.48	-.61	6.28	6.71	.96	.213
		7		983.85	958.7	5373.6	29.91	17.89	-589.0	9.33	.83	4.88	14.28	.54	.475
401- 501	LG2	5	0.0	61.70	394.6	56.9	-.48	18.51	17.1	.59	.34	.06	.93	.18	.042
		6		85.92	737.9	9212.2	-65.54	17.85	-3406.0	.81	.64	7.96	8.78	2.11	.203
		7		1198.96	3875.4	7365.5	-51.28	-14.15	-2120.6	11.37	3.34	6.35	18.36	1.42	.621
		5	3.0	62.77	1067.7	49.3	-.48	18.31	17.1	.60	.92	.04	1.52	.18	.065
		6		86.99	1385.8	6898.4	-61.46	17.59	-3406.0	.83	1.19	5.94	6.88	2.07	.220
		7		1200.84	3606.8	5513.8	-48.36	-11.47	-2120.6	11.38	2.94	4.78	16.99	1.39	.572
		5	6.1	63.85	1733.7	31.8	-.48	18.12	17.1	.61	1.69	.03	2.18	.18	.091
		6		88.87	2824.3	4128.2	-57.25	17.36	-3406.0	.86	1.74	4.07	5.25	2.83	.169
		7		1201.12	3838.2	3838.7	-45.34	-8.69	-2120.6	11.39	2.62	3.38	15.60	1.35	.529

FIGURE 26

FIGURE 3

SSI WAVE CARD

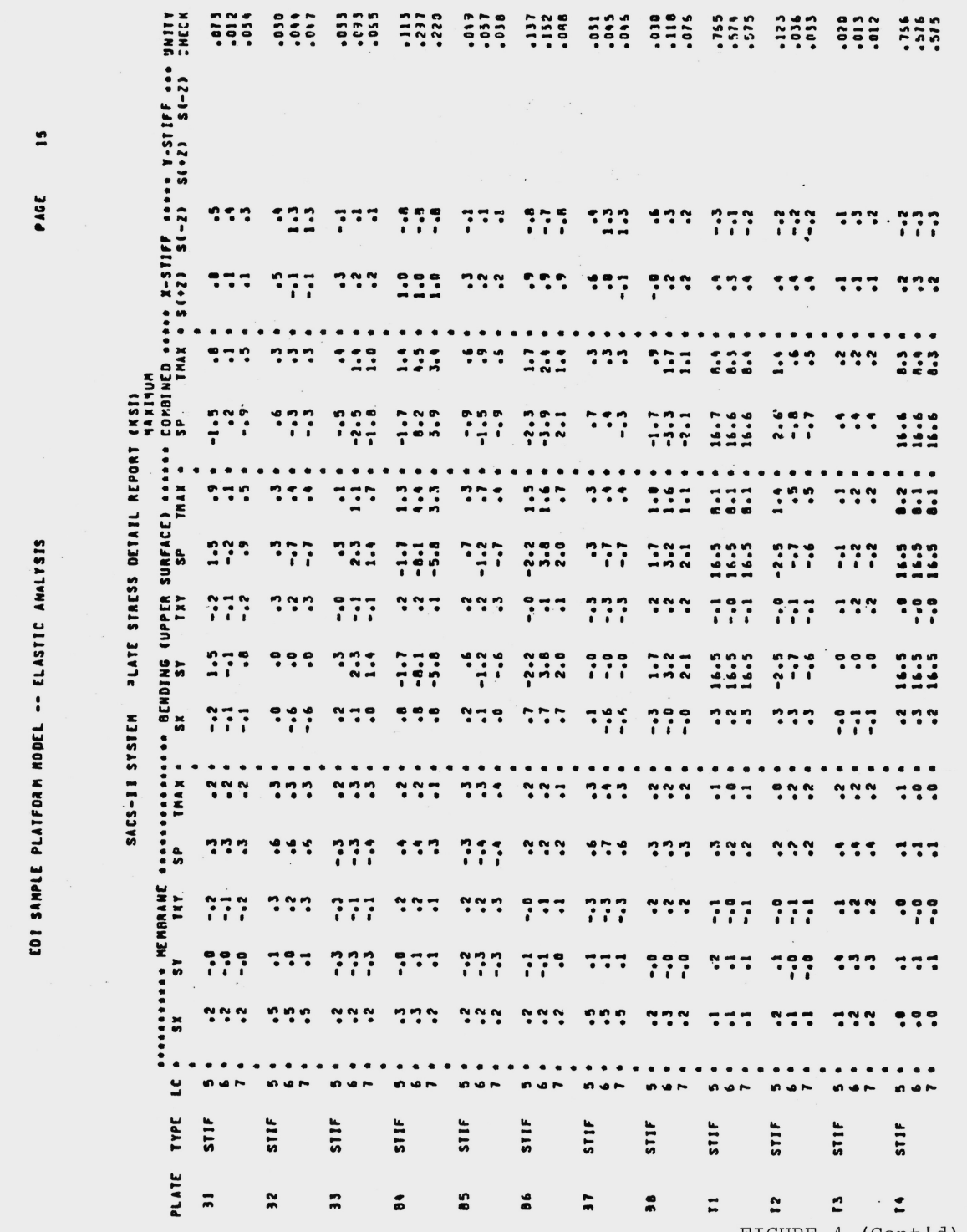
StruCAD*3D Ver. 3.12 VERIFICATION RUN Page 4

* * * Member Detail Report * * *

Member JA- JB	GRP ID	Load Case	Dist From End (ft)	Force Fx (Kips)	*Bending Moment* My (In-Kips)	Mz	*Shear Force* Fy (Kips)	Fz	Torsion Mx (In-Kips)	Axial Stress	Bending Stress Y	Z KSI	Comb. Stress	Shear Stress	Max Comb. Unity Check
301- 321	M2	1	.0	-38.39	-27.7	-193.2	.54	.12	2.9	-.94	-.16	-1.15	-2.10	.04	.131
		3		-39.16	-21.9	-204.0	.64	.06	3.2	-.96	-.13	-1.21	-2.18	.04	.135
		6		-109.10	15.6	-470.2	1.51	-.58	-2.5	-2.68	.09	-2.79	-5.48	.09	.348
		7		-110.66	7.6	-471.8	1.44	-.50	-2.4	-2.72	.05	-2.80	-5.53	.08	.273
		1	3.0	-38.39	-23.4	-173.6	.55	.11	2.9	-.94	-.14	-1.03	-1.99	.04	.127
		3		-39.16	-19.9	-180.8	.64	.06	3.2	-.96	-.12	-1.07	-2.04	.04	.130
		6		-108.66	-6.1	-414.1	1.61	-.62	-2.5	-2.67	-.04	-2.46	-5.13	.09	.353
		7		-110.23	-11.3	-418.0	1.55	-.55	-2.4	-2.71	-.07	-2.48	-5.20	.09	.263
		1	3.0	-38.39	-23.4	-173.6	.55	.11	2.9	-1.40	-.20	-1.48	-2.89	.05	.159
		3		-39.16	-19.9	-180.8	.64	.06	3.2	-1.42	-.17	-1.55	-2.98	.06	.164
		6		-108.66	-6.1	-414.1	1.61	-.62	-2.5	-3.95	-.05	-3.54	-7.49	.14	.442
		7		-110.23	-11.3	-418.0	1.55	-.55	-2.4	-4.01	-.10	-3.57	-7.58	.13	.328
		1	50.0	-38.39	4.7	193.1	.76	-.01	2.9	-1.40	.04	1.65	-3.05	.07	.165
		3		-39.16	11.7	181.5	.64	.06	3.2	-1.42	.10	1.55	-2.98	.06	.164
		6		-104.80	134.7	351.0	1.11	1.12	-2.5	-3.81	1.15	3.00	-7.03	.13	.417
		7		-106.42	127.0	375.0	1.28	1.05	-2.4	-3.87	1.08	3.20	-7.25	.13	.314
711- 701	M2	1	.0	190.01	524.7	60.1	-.37	-1.97	-33.8	4.67	3.12	.36	7.81	.20	.333
		3		197.57	548.2	54.5	-.25	-2.14	-34.3	4.86	3.26	.32	8.13	.21	.346
		6		281.78	946.3	-81.9	1.09	-3.97	-56.4	6.93	5.62	-.49	12.58	.37	.530
		7		288.06	960.0	-70.2	.93	-3.94	-58.3	7.09	5.70	-.42	12.81	.37	.405
		1	3.0	190.01	453.5	47.3	-.34	-1.98	-33.8	4.67	2.69	.28	7.38	.20	.317
		3		197.57	471.0	45.5	-.25	-2.14	-34.3	4.86	2.80	.27	7.67	.21	.329
		6		282.24	800.7	-45.3	.94	-4.12	-56.4	6.94	4.76	-.27	11.71	.38	.498
		7		288.52	815.3	-38.8	.81	-4.10	-58.3	7.10	4.84	-.23	11.95	.38	.382
		1	3.0	190.01	453.5	47.3	-.34	-1.98	-33.8	6.91	3.88	.40	10.81	.29	.464
		3		197.57	471.0	45.5	-.25	-2.14	-34.3	7.19	4.03	.39	11.23	.30	.483
		6		282.24	800.7	-45.3	.94	-4.12	-56.4	10.27	6.84	-.39	17.12	.55	.729
		7		288.52	815.3	-38.8	.81	-4.10	-58.3	10.50	6.97	-.33	17.47	.55	.559
		1	39.9	190.01	-470.8	-36.1	-.05	-2.21	-33.8	6.91	-4.02	-.31	10.95	.30	.469
		3		197.57	-477.9	-64.5	-.25	-2.14	-34.3	7.19	-4.08	-.55	11.31	.30	.485
		6		287.41	-754.0	-137.6	-1.36	-2.92	-56.4	10.46	-6.44	-1.18	17.01	.48	.727
		7		293.93	-780.4	-112.8	-1.16	-3.14	-58.3	10.69	-6.67	-.96	17.43	.49	.559
731- 711	CG3	1	.0	22.77	-1307.6	-692.1	11.04	-13.77	-164.2						
		3		20.70	-1174.6	-674.9	10.45	-14.65	-260.1						
		6		51.11	-4466.2	-1244.5	37.95	-1.24	1342.1						
		7		54.48	-4655.1	-1307.7	39.04	-1.63	1401.5						
		1	42.0	22.77	-8392.1	4624.8	10.07	-14.37	-164.2						
		3		20.70	-8565.2	4597.0	10.45	-14.65	-260.1						
		6		42.10	-15171.1	8437.7	.45	-41.23	1342.1						
		7		45.58	-15576.3	8776.8	.99	-41.75	1401.5						

FIGURE 4

SSI WAVE CARD (Cont'd)



EDI SAMPLE PLATFORM MODEL -- ELASTIC ANALYSIS PAGE 15

SACS-II SYSTEM PLATE STRESS DETAIL REPORT (KSI)

PLATE	TYPE	LC	MEMBRANE SX	MEMBRANE SY	MEMBRANE TXY	MEMBRANE SP	MEMBRANE TMAX	BENDING (UPPER SURFACE) SX	BENDING (UPPER SURFACE) SY	BENDING (UPPER SURFACE) TXY	BENDING (UPPER SURFACE) SP	BENDING (UPPER SURFACE) TMAX	COMBINED SP	COMBINED TMAX	X-STIFF S(+Z)	X-STIFF S(-Z)	Y-STIFF S(+Z)	Y-STIFF S(-Z)	UNITY CHECK
31	STIF	5	.2	-.0	-.2	.3	.2	-.2	1.5	-.2	1.5	.9	-1.5	.8	.8	.5			.073
		6	.2	-.0	-.1	.3	.2	-.1	-.1	-.1	-.2	.1	.2	.1	.1	.4			.012
		7	.2	-.0	-.2	.3	.2	-.1	.8	-.2	.9	.5	-.9	.5	.1	.3			.054
32	STIF	5	.5	.1	.3	.6	.3	.0	.0	.3	.3	.3	.6	.3	.5	.9			.030
		6	.5	.0	.2	.6	.3	-.6	.0	.2	-.7	.4	-.3	.3	-.1	1.3			.044
		7	.5	.1	.3	.6	.3	-.6	.0	.3	-.7	.4	-.3	.3	-.1	1.3			.017
33	STIF	5	.2	-.3	-.2	.4	.2	.2	.3	-.0	.3	.1	-.5	.4	.3	-.1			.013
		6	.2	-.3	-.1	.4	.3	.1	2.3	-.1	2.3	1.1	-2.5	1.4	.2	.1			.033
		7	.2	-.3	-.1	.4	.3	.0	1.4	-.1	1.4	.7	-1.8	1.0	.2	.1			.055
84	STIF	5	.3	-.0	.2	.4	.2	.8	-1.7	.2	-1.7	1.3	-1.7	1.4	1.0	-.8			.113
		6	.3	.1	.2	.4	.2	.8	-8.1	.2	-8.1	4.4	8.2	4.5	1.0	-.8			.237
		7	.2	.1	.1	.3	.1	.8	-5.8	.1	-5.8	3.3	5.9	3.4	1.0	-.8			.220
85	STIF	5	.2	-.2	.2	.3	.2	.2	.6	.2	.7	.3	-.9	.6	.3	-.1			.019
		6	.2	-.3	.2	.3	.2	.1	-1.2	.2	-1.2	.7	-1.5	.9	.2	.1			.057
		7	.2	-.3	.3	.4	.2	.0	-.6	.3	-.7	.4	-.9	.5	.2	.1			.018
86	STIF	5	.2	-.1	-.0	.2	.2	.7	-2.2	-.0	-2.2	1.5	-2.3	1.7	.9	-.8			.137
		6	.2	-.1	.1	.2	.2	.7	3.8	.1	3.8	1.6	-3.9	2.4	.9	-.7			.132
		7	.2	.0	.1	.2	.1	.7	2.0	.1	2.0	.7	2.1	1.4	.9	-.8			.048
87	STIF	5	.5	.1	-.3	.6	.3	.1	-.0	-.3	.3	.3	.7	.3	.6	.9			.031
		6	.5	.1	-.3	.7	.4	-.6	-.0	-.3	-.7	.4	.4	.3	.8	1.3			.045
		7	.5	.1	-.3	.6	.3	-.6	-.0	-.3	-.7	.4	-.3	.3	-.1	1.3			.045
88	STIF	5	.2	-.0	.2	.3	.2	-.3	1.7	.2	1.7	1.0	-1.7	.9	-.0	.6			.030
		6	.3	-.0	.2	.3	.2	-.0	3.2	.2	3.2	1.6	-3.3	1.7	.2	.3			.110
		7	.2	-.0	.2	.3	.2	-.0	2.1	.2	2.1	1.1	-2.1	1.1	.2	.2			.075
11	STIF	5	.1	.2	-.1	.3	.1	.3	16.5	-.1	16.5	8.1	16.7	8.4	.9	-.3			.755
		6	.1	.1	-.0	.2	.0	.2	16.5	-.0	16.5	8.1	16.6	8.3	.3	-.1			.574
		7	.1	.1	-.1	.2	.1	.3	16.5	-.1	16.5	8.1	16.6	8.4	.4	-.2			.575
12	STIF	5	.2	-.1	-.0	.2	.0	.3	-2.5	-.0	-2.5	1.4	2.6	1.4	.4	-.2			.123
		6	.1	-.0	-.1	.2	.2	.3	-.7	-.1	-.7	.5	-.8	.6	.4	-.2			.036
		7	.1	-.0	-.1	.2	.2	.3	-.6	-.1	-.6	.5	-.7	.5	.4	-.2			.033
13	STIF	5	.1	.4	.1	.4	.2	-.0	.0	.1	-.1	.1	.4	.2	.1	.1			.020
		6	.2	.3	.2	.4	.2	-.1	.0	.2	-.2	.2	.4	.2	.1	.3			.013
		7	.2	.3	.2	.4	.2	-.1	.0	.2	-.2	.2	.4	.2	.1	.2			.012
14	STIF	5	.0	.1	.0	.1	.1	.2	16.5	.0	16.5	8.2	16.6	8.3	.2	-.2			.756
		6	.0	.1	-.0	.1	.0	.3	16.5	-.0	16.5	8.1	16.6	8.4	.3	-.3			.576
		7	.0	.1	-.0	.1	.0	.2	16.5	-.0	16.5	8.1	16.6	8.3	.2	-.3			.575

3-5-31

FIGURE 4 (Cont'd)

Interestingly, EDI never registered its copyrights in any of the 23 modules comprising the SACS program, preferring to rely instead on trade secret protection. It did, however, obtain four registrations covering the user manuals for three of the 23 modules, which contained detailed verbal descriptions and pictorial representations of 51 of the more than 200 input formats used by the SACS program modules and of various output formats.

In 1986, SSI released a competing structural analysis program called StruCAD that borrowed heavily from the SACS user interface, adopting a similar, though not identical, 80-column format. Unlike the earlier Synercom case, however, StruCAD was *not* completely compatible with SACS. For example, manual individual data cards completed for use in SACS would require some, but not extensive, modification before they could be used with StruCAD, and StruCAD required dozens of input formats completely different from those found in the SACS program.

EDI brought suit against SSI, claiming that 56 of EDI's input formats had been copied from its copyrighted user manuals into SSI's user manual and into the StruCAD user interface, including its help screens. EDI also alleged copying of its output report formats. (Figures 5 and 6 show samples of EDI's Member Detail output report and SSI's corresponding Member Detail report; Figures 7 and 8 show samples of EDI's Plate Stress output report and SSI's corresponding Plate Stress report). Unlike the Synercom case, in which only nine input formats were alleged to have been copied and Synercom asserted copyright protection over each of such formats individually, EDI did not claim protection for any of its individual input formats and output reports. Instead, it contended that the sequence and organization of its formats and reports were, as a whole, copyrightable. SSI defended on the grounds that the input formats were not copyrightable and that, because EDI had allegedly copied many of the formats from Synercom, it could not now assert a proprietary interest in them. After a four-day bench trial, the district court ruled that the input and output formats were not copyrightable, but held that SSI had infringed EDI's copyrights in its user manuals.

2. The Fifth Circuit's Decision

On appeal, the Fifth Circuit reversed the ruling that the input and output formats were not copyrightable. As in the case of the Tenth Circuit's decision in the Gates Rubber case, the Fifth Circuit formulated an infringement test based upon two separate inquiries:

1) "Factual Copying": Whether the defendant, as a factual matter, "actually used the copyrighted material to create his own work."[407] Citing Professor Latman and the Gates Rubber case, the Court ruled that "[c]opying as a factual matter typically may be inferred from proof of access to the copyrighted work and 'probative similarity.'[408]

---

[407] 26 F.3d at 1340.

[408] Id. ("Professor Latman distinguishes between 'probative similarity,' which relates to factual copying, and 'substantial similarity,' which relates to actionable copying.") (citing 3 Melville B. & David Nimmer, Nimmer on Copyright § 13.01[B] (1993)).

EDI MEMBER DETAIL REPORT

WAVE DESCRIPTION CARD (Cont'd.)

Column	Format	Description
59-64	F6.2	Step size defining the crest position for load calculation in distance, degrees, or seconds. The distance default units are: US(FT), Metric(M), SI(M).
65-68	I4	Number of wave steps to be made. For input number of 'N', loads are calculated @ 'N+1' positions.
69-70	A2	Enter MM for maximum overturning moment NM for minimum overturning moment MS for maximum base shear NS for minimum base shear MU for maximum upward force MD for maximum downward force
71-72	I2	Maximum number of segments into which members should be divided for load calculations. Maximum segments allowed is 19. (Defaults to 10).
73-74	I2	Minimum number of segments into which members should be divided for load calculations. (Defaults to 1).
76	I1	Printout option. Enter 3 for maximum output. If left blank, no wave table is printed.
77-78	I2	If wave type is 'STRE', enter desired order of stream function. Allowable maximum is 22. Default is 7.

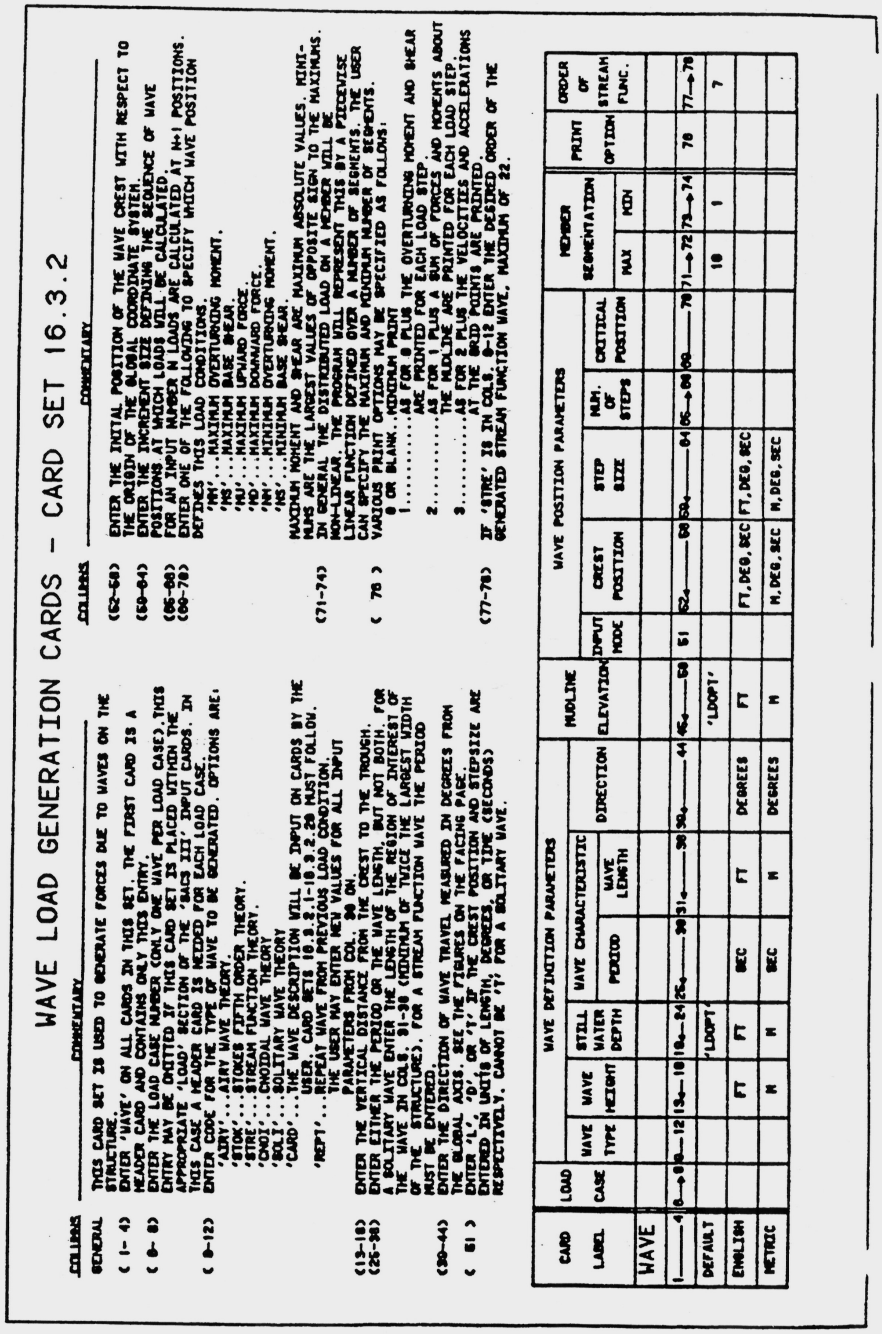
A-2-88

001278 SDE2

FIGURE 5

STRUCTURAL SOFTWARE, INC.

SSI MEMBER DETAIL REPORT



WAVE LOAD GENERATION CARDS - CARD SET 16.3.2

COLUMNS	COMMENTARY	COLUMNS	COMMENTARY
GENERAL	THIS CARD SET IS USED TO GENERATE FORCES DUE TO WAVES ON THE STRUCTURE.	(62-64)	ENTER THE INITIAL POSITION OF THE WAVE CREST WITH RESPECT TO THE ORIGIN OF THE GLOBAL COORDINATE SYSTEM.
(1- 4)	ENTER 'WAVE' ON ALL CARDS IN THIS SET. THE FIRST CARD IS A HEADER CARD AND CONTAINS ONLY THIS ENTRY.	(69-64)	ENTER THE INCREMENT SIZE DEFINING THE SEQUENCE OF WAVE POSITIONS AT WHICH LOADS WILL BE CALCULATED.
(6- 8)	ENTER THE LOAD CASE NUMBER (ONLY ONE WAVE PER LOAD CASE). THIS ENTRY MAY BE OMITTED IF THIS CARD SET IS PLACED WITHIN THE APPROPRIATE 'LOAD' SECTION OF THE 'SAS III' INPUT CARDS. IN THIS CASE A HEADER CARD IS NEEDED FOR EACH LOAD CASE.	(65-66)	FOR AN INPUT NUMBER N LOADS ARE CALCULATED AT N+1 POSITIONS.
		(66-70)	ENTER ONE OF THE FOLLOWING TO SPECIFY WHICH WAVE POSITION DEFINES THIS LOAD CONDITIONS.
(9-12)	ENTER CODE FOR THE TYPE OF WAVE TO BE GENERATED. OPTIONS ARE:		'MV'...MAXIMUM OVERTURNING MOMENT.
	'AIRY'...AIRY WAVE THEORY.		'MS'...MAXIMUM BASE SHEAR.
	'STOK'...STOKES FIFTH ORDER THEORY.		'MU'...MAXIMUM UPWARD FORCE.
	'STRE'...STREAM FUNCTION THEORY.		'MD'...MAXIMUM DOWNWARD FORCE.
	'CNOI'...CNOIDAL WAVE THEORY		'MM'...MAXIMUM OVERTURNING MOMENT.
	'SOLI'...SOLITARY WAVE THEORY		'MS'...MINIMUM BASE SHEAR.
	'CARD'...THE WAVE DESCRIPTION WILL BE INPUT ON CARDS BY THE USER. CARD SETS 16.3.2.1-16.3.2.20 MUST FOLLOW.		MAXIMUM MOMENT AND SHEAR ARE MAXIMUM ABSOLUTE VALUES. MINIMUMS ARE THE LARGEST VALUES OF OPPOSITE SIGN TO THE MAXIMUMS.
	'REPT'...REPEAT WAVE FROM PREVIOUS LOAD CONDITION. THE USER MAY ENTER NEW VALUES FOR ALL INPUT PARAMETERS FROM COL. 20 ON.	(71-74)	IN GENERAL, THE DISTRIBUTED LOAD ON A MEMBER WILL BE NON-LINEAR. THE PROGRAM WILL REPRESENT THIS BY A PIECEWISE LINEAR FUNCTION. DEFINED OVER A NUMBER OF SEGMENTS. THE USER CAN SPECIFY THE MAXIMUM AND MINIMUM NUMBER OF SEGMENTS.
(13-18)	ENTER THE VERTICAL DISTANCE FROM THE CREST TO THE TROUGH.	(75)	VARIOUS PRINT OPTIONS MAY BE SPECIFIED AS FOLLOWS:
(25-30)	ENTER EITHER THE PERIOD OR THE WAVE LENGTH, BUT NOT BOTH. FOR A SOLITARY WAVE ENTER THE LENGTH OF THE REGION OF INTEREST OF THE WAVE IN COLS. 31-36 (MINIMUM OF TWICE THE LARGEST WIDTH OF THE STRUCTURE). FOR A STREAM FUNCTION WAVE THE PERIOD MUST BE ENTERED.		0 OR BLANK...MINIMUM PRINT
			1.....AS FOR 0 PLUS THE OVERTURNING MOMENT AND SHEAR ARE PRINTED FOR EACH LOAD STEP.
			2.....AS FOR 1 PLUS A SUM OF FORCES AND MOMENTS ABOUT THE MUDLINE ARE PRINTED FOR EACH LOAD STEP.
(30-44)	ENTER THE DIRECTION OF WAVE TRAVEL MEASURED IN DEGREES FROM THE GLOBAL AXIS. SEE THE FIGURES ON THE FACING PAGE.		3.....AS FOR 2 PLUS THE VELOCITIES AND ACCELERATIONS AT THE GRID POINTS ARE PRINTED
(81)	ENTER 'L', 'D', OR 'I'. IF THE CREST POSITION AND STEPSIZE ARE ENTERED IN UNITS OF LENGTH, DEGREES, OR TIME (SECONDS) RESPECTIVELY. CANNOT BE 'I' FOR A SOLITARY WAVE.	(77-78)	IF 'STRE' IS IN COLS. 9-12 ENTER THE DESIRED ORDER OF THE GENERATED STREAM FUNCTION WAVE, MAXIMUM OF 22.

CARD LABEL	LOAD CASE	WAVE DEFINITION PARAMETERS: WAVE TYPE	WAVE HEIGHT	STILL WATER DEPTH	WAVE CHARACTERISTIC: PERIOD	WAVE CHARACTERISTIC: WAVE LENGTH	DIRECTION	MUDLINE ELEVATION	WAVE POSITION PARAMETERS: INPUT MODE	CREST POSITION	STEP SIZE	NUM. OF STEPS	CRITICAL POSITION	MEMBER SEGMENTATION: MAX	MEMBER SEGMENTATION: MIN	PRINT OPTION	ORDER OF STREAM FUNC.
WAVE																	
1——4	8→8	12	13→18	24→30	31→36	30→44	45→50	51	52→58	59→64	65→70	71→72	73→74	75	77→78		
DEFAULT				'LOOP1'				'LOOP1'						10	1	7	
ENGLISH			FT	FT	SEC	FT	DEGREES	FT		FT, DEG, SEC	FT, DEG, SEC						
METRIC			M	M	SEC	M	DEGREES	M		M, DEG, SEC	M, DEG, SEC						

2-4-35

REV-006-84

FIGURE 6

EDI PLATE STRESS REPORT

WAVE DESCRIPTION CARD

The loads on the structure due to wave with or without current can be generated using this card. At present the hydrodynamic loads can be generated using either Stoke's 5th order wave theory or Stream function wave theory. The hydrodynamic loads are calculated using Morrison's equation.

Column	Format	Description
1-4	A4	Card type designation. Enter WAVE.
9-12	A4	Enter 'STOK' for Stokes' 5th order theory 'STRE' for Dean's Stream function 'AIRY' for Airy wave theory 'GRID' Grid data input 'REPT' Use previous wave
13-18	F6.2	Wave height, the vertical distance from trough to crest. The default units are: US(FT), Metric(M), SI(M).
19-24	F6.2	Water depth. Defaults to the definition on LDOPT card. The default units are: US(FT), Metric(M), SI(M).
25-30	F6.2	Wave period in seconds.
39-44	F6.2	Wave direction in degrees measured from the global X-axis, using right hand system.
45-50	F6.2	Mudline elevation. Defaults to the definition on LDOPT card. The default units are: US(FT), Metric(M), SI(M).
51	A1	Enter L, D or T to indicate that crest position and step size is in either distance, degrees or seconds, respectively. The distance default units are: US(FT), Metric(M), SI(M).
52-58	F7.2	Initial crest position in distance, degrees, or seconds, as appropriate. The distance is measured from the global origin to the wave crest, positive in the wave direction. The distance default units are: US(FT), Metric(M), SI(M).

A-2-87

001278

STRUCTURAL SOFTWARE, INC.

FIGURE 7

SSI PLATE STRESS REPORT

StruCAD*3D Ver. 3.12 VERIFICATION RUN Page 35

* * * Report Of Plate Stresses (KSI) * * *

PNAME	NO.	LC	Membrane Only Sx	Sy	Txy	Sp	Tmax	Bending Only (Top) Sx	Sy	Txy	Sp	Tmax	Combined Sp1	Sp2	Tmax	Von Mises	Unity Ratio	CRT PNT
PL2	2	1	.01	.11	-2.54	2.60	2.54	-.01	-.03	.01	-.04	.02	2.63	-2.47	2.55	4.42	.205	BOT
		3	.00	.09	-2.53	2.58	2.53	-.01	-.03	.01	-.04	.02	2.61	-2.47	2.54	4.40	.204	BOT
		6	-.06	-.10	-6.12	-6.20	6.12	-.02	-.06	.02	-.07	.03	6.11	-6.19	6.15	10.64	.493	BOT
		7	-.05	-.07	-6.29	-6.35	6.29	-.02	-.06	.02	-.07	.03	6.29	-6.33	6.31	10.93	.380	BOT
PL3	3	1	-.50	-1.72	-.33	-1.80	.70	-.07	-.09	-.01	-.10	.02	-.48	-1.90	.71	1.71	.079	TOP
		3	-.50	-1.74	-.29	-1.80	.68	-.07	-.09	-.01	-.10	.02	-.50	-1.89	.70	1.70	.079	TOP
		6	-1.39	-4.94	2.70	-6.40	3.23	-.16	-.20	-.03	-.21	.03	-.13	-6.57	3.22	6.51	.301	TOP
		7	-1.42	-5.01	2.59	-6.37	3.16	-.17	-.20	-.03	-.22	.03	-.26	-6.54	3.14	6.42	.223	TOP

FIGURE 3

2) Legally Actionable Copying: Whether "there is substantial similarity between the two works."[409]

The Fifth Circuit's two-step analysis, like that of the Tenth Circuit in Gates Rubber, applies similarity *twice*. First, "probative similarity" in the works is determined to decide the factual question of whether the defendant copied from the plaintiff's work. If so, the fact finder must determine whether there is "substantial similarity" between the works at issue sufficient to find *unlawful* copying. Because factual copying was not disputed in the case, the Court focused its analysis on the question of whether the allegedly copied input and output formats were copyrightable.

(a) Some Preliminary Observations

Before turning to the Fifth Circuit's analysis of the copyrightability question, two interesting observations may be made. *First*, SSI argued that the Court should not analyze the input formats as nonliteral elements of a computer program, because that analysis depends on the existence of a copyright on the underlying computer program and, as previously noted, EDI had not registered its copyrights on its computer programs. The Court rejected this argument: "It makes no difference to the formats' copyrightability whether we analyze them as springing from a computer program or from a user manual."[410] Thus, the Engineering Dynamics case suggests that a plaintiff can base a claim for copyright infringement of nonliteral elements of the input formats or output formats of the program on the copyright in the *user manual* describing those formats. The case also suggests an interesting way in which to use a combination of trade secret protection for a computer program with copyright protection for an accompanying user manual.

Second, SSI relied heavily for its defense on the earlier Synercom case. Both SSI and the district court read the Fifth Circuit's decision in Plains Cotton Cooperative v. Goodpasture Computer Serv.[411] to have adopted a broad reading of Synercom to hold that nonliteral elements of computer programs are not copyrightable. The Engineering Dynamics opinion states, however, that the Fifth Circuit's decision in Kepner-Tregoe, Inc. v. Leadership Software Inc.[412] established that Plains Cotton did not adopt Synercom, at least insofar as Synercom is read to stand for the broad proposition that nonliteral elements of computer programs are not copyrightable. Synercom did not, in fact, adopt such a broad, blanket rule against the copyrightability of all nonliteral elements of a computer program. Rather, it held that the particular formats of data on 80-column cards at issue were merged with their underlying idea and were therefore not copyrightable. Nevertheless, both the Fifth Circuit's disagreement with

---

[409] Id. at 1341.

[410] Id. at 1342.

[411] 807 F.2d 1256 (5th Cir.), cert. denied, 484 U.S. 821 (1987).

[412] 12 F.3d 527 (5th Cir. 1994). The Kepner-Tregoe case embraced the "general, noncontroversial proposition that nonliteral aspects of copyrighted works – like structure, sequence and organization – may be protected under copyright law." Id. at 536 n.20.

Synercom as broadly characterized and the outcome of the Engineering Dynamics case itself suggest that the Synercom case is probably no longer good law in the Fifth Circuit.[413]

(b) Adoption of the Abstraction/Filtration/Comparison Test

The Fifth Circuit began its analysis with a general endorsement of the abstraction/filtration/comparison analysis elucidated by the Tenth Circuit in Gates Rubber, although the Court cautioned that "[p]rotectable originality can manifest itself in many ways, so the analytic approach may need to be varied to accommodate each case's facts."[414]

(i) Abstraction. The Fifth Circuit opined that application of the abstraction step did not pose a problem in this case because the plaintiff was seeking protection for its input and output formats "not individually but *en masse*. It is thus unnecessary at this stage to decide whether each individual input format card or output format report represents an idea or an expression."[415] With very little analysis, the Court concluded that such formats, taken as a whole, qualified as "expression" rather than "idea":

> The purpose of the SACS input formats is to mediate between the user and the program, identifying what information is essential and how it must be ordered to

---

[413] The Engineering Dynamics case also distinguishes Synercom on the ground that in Synercom the plaintiff sought copyright protection for individual input formats, whereas EDI was arguing that several dozen input formats *taken together* could form a copyrightable work "because they represent but one of many ways of expressing a mode of computerized structural analysis." Engineering Dynamics, 26 F.3d at 1342. Despite this effort to distinguish Synercom, it nevertheless appears that the Fifth Circuit has now virtually disavowed Synercom.

[414] 26 F.3d at 1343. In a case decided only five months earlier, the Fifth Circuit had made reference to an "abstraction-filtration-comparison" test, citing the Second Circuit's Altai case, without explicitly adopting that test:

> To determine the scope of copyright protection in a close case, a court may have to filter out ideas, processes, facts, idea/expression mergers, and other unprotectable elements of plaintiff's copyrighted materials to ascertain whether the defendant infringed protectable elements of those materials. ... Although there is no evidence that the district court undertook a rigorous "abstraction-filtration-comparison" analysis of the sort approved by courts for sophisticated treatment of copyright cases, such an analysis was not absolutely necessary here. The district court carefully juxtaposed selections from K-T's [copyrighted written materials on leadership training] with selections from the [defendant's management] program, thereby demonstrating a damning similarity – nay identity – of organization and language.

Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527, 533-34 (5th Cir. 1994) (citation to Altai omitted).

[415] Engineering Dynamics, 26 F.3d at 1344.

> make the program work. The output formats structure the results of calculations performed by the program informatively for the user.[416]

Thus, in much the same way that Judge Keeton approached the copyrightability of the commands of the Lotus 1-2-3 user interface in Borland, it appears that the fact that the input and output formats at issue could be characterized as conveying information to the user was of itself sufficient for the Fifth Circuit to conclude that such formats constituted expression rather than idea.[417] The Court also noted that "there are numerous ways in which either input or output formats could have been structured in order achieve the program's purpose."[418]

(ii) Filtration. The Court explicitly noted that filtration must be applied to *each* level of abstraction that has been identified in the Abstraction step. The Court noted the following uncopyrightable elements that must be filtered from each particular level: ideas, processes, methods or scientific discoveries, facts, information in the public domain, and scenes a faire ("*i.e.*, expressions that are standard, stock or common to a particular subject matter or are dictated by external factors").[419]

The Court rejected SSI's argument that SSI's formats were akin to uncopyrightable facts because they were merely shorthand for a common engineering formula: "What appears on EDI's input and output formats, however, are not any kind of formulas or 'facts' as such, but organized, descriptive tables for entry of data on which the computer will perform necessary calculations."[420] The Court noted that there were many other ways to organize such tables.

The Court also rejected SSI's analogy of the formats to an uncopyrightable template, process or method, again based largely on the fact that the formats conveyed information to the xuser:

> The question is whether the utilitarian function of the input formats, which ultimately act like switches in the electrical circuits of the program, outweigh their expressive purpose so as to preclude copyright protection. On balance, we believe they do not. EDI's input formats as a whole convey substantial information regarding what data the user needs to gather and how they should be organized for the program to run properly. One of EDI's trial witnesses testified

---

[416] Id.

[417] The Fifth Circuit appears to have been influenced by Judge Keeton's pragmatic approach to the necessity for copyright protection in resolving the issue of copyrightability, for the Fifth Circuit quoted Judge Keeton's observation that "if a best-selling program's interface were not copyrightable, competitors would be free to emulate the popular interface exactly so long as the underlying programs were not substantially similar. This cannot be the law." Id.

[418] Id.

[419] Id.

[420] Id. at 1345.

> that the interface "imparts knowledge" by telling the user which data to collect as well as the order of collection.[421]

The Court observed that "generally, functional interfaces that directly teach or guide the user's independent decisions are more expressive than functional interfaces that lack these qualities. Although the degree of interaction may not be as high as that present in Lotus, overall, EDI has proved original expressive content in the selection, sequence and coordination of inputs."[422]

The Court next rejected SSI's argument that the formats at issue were merely a garden variety arrangement of engineering data required in any structural analysis program that did not comprise sufficient originality under the Feist case[423] to be copyrightable as a compilation. The Court doubted whether Feist should apply at all, since it had earlier rejected SSI's argument that the formats were akin to "facts," but further ruled that there was sufficient originality in EDI's particular selection and arrangement of formats, again pointing to the fact that other, dissimilar structural engineering programs were available in the market.[424]

Finally, the Court considered SSI's argument that the EDI formats were akin to scenes a faire because they were dictated by the externality of "the nature of the offshore structural engineering marketplace."[425] The Court found this argument to have some force, for it ruled that "[o]n remand, the district court must consider whether or to what extent industry demand and practice in the offshore engineering market dictated the SACS IV input and output formats."[426] The Court specifically noted that some of the allegedly infringed cards may be so generic "*e.g.*, a 'header' or an 'end' card, that they lack that minimal degree of creativity required for copyright protection."[427] Similarly, if other cards were almost wholly derived from the input formats developed by Synercom many years earlier, they would lack sufficient originality to be copyrightable and would have to be filtered out of the analysis.

(iii) Comparison. The Fifth Circuit remanded the case to the district court for a determination of whether SSI's and EDI's input formats and output reports were, taken as a whole, substantially similar, after filtering out any unprotectable cards. The Court provided some interesting guidance to the district court on how such comparison should be conducted. In particular, similar to the Ninth Circuit's test in the Apple v. Microsoft case, the

---

[421] Id. at 1346.

[422] Id.

[423] Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340 (1991) (holding that telephone book white page listings were not copyrightable as a compilation because there was no expressive selection, coordination, or arrangement in a mere alphabetical listing of all names of residents in an area).

[424] See Engineering Dynamics, 26 F.3d at 1346.

[425] Id.

[426] Id. at 1347.

[427] Id.

Court suggested that comparison should be conducted at the *individual* card level first, then at the "works as a whole" level:

> In this case, it is probably advisable for the court first to determine whether variations in the registered and copyrightable format cards adopted by StruCAD render the cards noninfringing elements of the larger work at the individual card level. Then the court may determine whether the subset of StruCAD cards that are individually substantially similar to their counterparts in SACS, are, taken together, so substantially similar to EDI's copyrighted work *or a part thereof* as to constitute infringement.[428]

The second piece of guidance provided by the Fifth Circuit for the court on remand was also similar to the Ninth Circuit's approach in Apple v. Microsoft. The Fifth Circuit noted that the standard of similarity to be applied in the comparison of the works as a whole (or, equivalently, the scope of protection to be afforded to the copyrighted work) can vary from case to case:

> Another proposition to bear in mind is that the scope of protection afforded by a copyright is not constant across all literary works. ... The same cautious approach to protection is appropriate for computer user interfaces. To the extent that they are highly functional, or, like the output formats in this case, to the extent that they contain highly standardized technical information, they may lie very near the line of uncopyrightability.[429]

Although the Fifth Circuit did not speak explicitly in terms of a "virtual identity" standard, the Court cited Judge Keeton's reference to a "sliding scale" of protection and to one of Judge Walker's opinions in the Apple v. Microsoft case in which he observed that only a "virtually identical" copy will be actionable if technical or conceptual constraints limit the available ways to express an idea.[430] Thus, although the precise contours of the way in which the Fifth Circuit intends for the standard of similarity to be adjusted from case to case are not spelled out, the Engineering Dynamics case suggests that the Fifth Circuit's approach may not be very different from the Ninth Circuit's approach in the Apple v. Microsoft case of using a continuum between "broad" and "thin" protection based upon the nature and scope of expression contained in the allegedly infringed work after the filtration step.

3. Clarification of the Fifth Circuit's Opinion

In August of 1994, the defendants filed a petition for rehearing and rehearing en banc, and for clarification of the Fifth Circuit's opinion. Although the Court rejected the petition for

---

[428] Id. In response to a motion for rehearing and rehearing en banc, the Fifth Circuit issued a supplemental opinion deleting the italicized language of this passage. See subsection 3 below.

[429] Id. at 1348.

[430] See id.

rehearing and rehearing on banc, on February 16, 1995, the Fifth Circuit issued a supplemental opinion to clarify the following points raised by the defendant concerning the original panel opinion:

First, both the petition for rehearing and amicus petitions in support of rehearing suggested that the initial panel opinion held that EDI's user formats are protected by copyright law because there "are numerous ways the input formats could be organized."[431] The Fifth Circuit responded that this characterization was "an overly simplistic view of the opinion. ... The panel did not say that in any case involving user interface the fact that the 'author' has selected from among possible formats is dispositive."[432] Rather, the Fifth Circuit had merely held that the input/output formats at issue in the case were more than a blank form, and on remand the district court must inquire into whether EDI exercised any judgment in formulating the input cards or merely reflected the industry standards and laws of engineering.[433]

Second, in response to concerns raised pointedly by amicus petitions, the Fifth Circuit clarified that its opinion "cannot properly be read to extend to the manufacturing of computer hardware so as to deter achieving compatibility with other models or to the practice employed by users of programs of analyzing application programs to 'read' the file formats of other programs."[434] For legal support, the Court cited the statement in the Gates Rubber opinion that the scenes a faire doctrine excludes protection of aspects of a work serving a functional purpose, and cited the following as examples of such functional aspects in the area of computer programs: hardware standards and mechanical specifications, "software standards and compatibility requirements," computer manufacturer design standards, target industry practices and demands, and computer industry programming practices.[435]

The Fifth Circuit's preceding comments are confusing. The Court did not elaborate on what it meant by "compatibility requirements," which would seem to have been the reason for the adoption of at least the same input formats in the defendants' program, so that the defendants' program could read the input files written for use with EDI's program. Thus, it is unclear why this cross-application input format compatibility did not fall within the reach of the Fifth Circuit's statement that its opinion should not be read to prohibit "analyzing application programs to 'read' the file formats of other programs."

Third, the Fifth Circuit admitted that its original opinion contained some ambiguity in the passage quoted in subsection 2(b)(iii) above – in which the Court stated that a similarity comparison should first be made at the individual card level, then of the input formats as a whole – concerning what the benchmark for substantial similarity is to be (individual elements or the

---

[431] Engineering Dynamics, Inc. v. Structural Software, Inc., 46 F.3d 408,409 (5th Cir. 1995).

[432] Id.

[433] Id. at 410.

[434] Id.

[435] Id.

compilation en masse). Accordingly, the Court ordered that the words "or a part thereof" be deleted from the quoted passage and the following sentence added:

> Of course SSI may not replicate component parts of EDI's protected work with impunity; substantial similarity may be measured by comparing the products as a whole, but the more exact a duplication of constituent pieces of a work the less overall similarity that may be required.[436]

Although this passage seems no less unclear than the previous passage, the Fifth Circuit seems to be saying that substantial similarity is to be judged in the case of a compilation based on similarity in the compilation as a whole. However, the amount of such overall similarity required for a finding of infringement will increase or decrease based on the amount of similarity in individual component pieces of the compilation. Presumably, if the defendants had copied some of the individual input formats exactly or nearly exactly, then overall similarity could more readily be found even if only a few such individual formats were copied. Conversely, if none of the individual formats were identical, then presumably similarities in a larger number of them would be required to justify a finding of overall substantial similarity. The Fifth Circuit does not address, however, the logical conundrum of why – if individual formats are not copyrightable, but only the compilation of them is copyrightable – the similarity of individual formats should be determinative of the degree of overall similarity to be required for a finding of infringement.

J. THE MITEL CASE

1. The District Court's Decision

A 1995 case from Colorado, Mitel, Inc. v. Iqtel, Inc.,[437] raised the issue of whether instruction sets used to program telephone call controllers are copyrightable. The plaintiff Mitel, Inc. manufactured and sold a telephone call controller, which is a device (generally resident in the central office of the telephone system) that permits a telephone user to use "1 plus" dialing to access its long-distance carrier of choice without having to dial a special access code. The call controller is also capable of providing functions that enhance the telephone service, such as speed dialing, tone-to-pulse conversion, automatic redial, and call detail recording.

Each of the functions of the call controller must be programmed into the call controller using a series of commands or "codes" entered through a telephone keypad or a lap-top computer connected to the phone line. The codes are three and four digit numbers or letters that specify the particular function of the call controller. Mitel's codes were described in its instructions manuals for the call controller, and Mitel argued that the copyright in the manuals covered the command codes described therein. The Mitel manual called the first three digits of a code the "register." The first number in the register referenced the line accessed. The next two digits accessed the particular function. The last number in the code, called the "description," represented a value attached to the particular function (such as the time between trunk release

---

[436] Id.

[437] 896 F. Supp. 1050 (D. Colo. 1995).

and next attempt). Each function in the Mitel controller had a separate registration code and description, which together Mitel called an "instruction set."

The defendant Iqtel, Inc. manufactured a competing call controller designed to be programmed in the same manner as Mitel's call controller. Iqtel determined that, in order for its call controller to be competitive in the marketplace, it had to be compatible with the Mitel call controller because Mitel then controlled between 75% and 90% of the call controller market. The Iqtel controller also used a four digit code, with some minor differences in format: the first two digits in the Iqtel code accessed the function and the third digit accessed the line. As in the case of the Mitel controller, the fourth digit of the Iqtel code represented a value, and the Iqtel values were identical to the Mitel values. The Iqtel controller also contained a translator that permitted the technician to enter Mitel codes into the Iqtel call controller to access the Iqtel functions. The designer of Iqtel's call controller admitted that he copied the registers and descriptions from Mitel's instruction manual in developing the translator.

Because the Mitel call controller could be programmed from a lap-top computer, many of the long distance carriers had written "macros" containing Mitel command codes that could be entered directly into the call controller. These macros contained the individual carrier's specifications for each function, which could be downloaded simultaneously, rather than input manually one at a time, to significantly reduce installation time.

Mitel did not claim that its controller command codes constituted either source code or object code of a computer program. Nor were the Mitel instruction sets part of the software contained in the Mitel controller. Rather, the instruction sets constituted command inputs into the call controller that caused the software within the controller to perform certain functions. Thus, the case raised the question whether an instruction set per se is copyrightable.

Mitel sought a preliminary injunction against Iqtel. Iqtel contended that the command codes were an uncopyrightable method of operation. The district court denied the preliminary injunction, concluding that the Mitel command codes were not protectable components of its copyrighted instruction manuals. Specifically, the court held that the command codes "are simply a procedure, process, system, and method of operation by which the customer can match the call controller functions to the long-distance carriers' technical needs and end-user's choices. Without the command codes the function would not occur and the results would not be achieved."[438] The court analogized the command codes to the replacement parts numbers held uncopyrightable in Toro Co. v. R&R Prods. Co.[439]

The court further noted that it would arrive at the same conclusion applying the abstraction/filtration/comparison test of Gates Rubber. Under that test, the court held that, even considering the command codes to be part of the computer program in the Mitel call controller, the codes must be filtered out as an uncopyrightable means to access or operate the controller.

---

[438] Id. at 1055.

[439] 787 F.2d 1208 (8th Cir. 1986) (defendant's use of a catalog which indexed various replacement parts by the original manufacturer's part number and name held not to violate a claimed copyright in the plaintiff's replacement parts numbering system).

"The command codes in this case simply act as a key to unlock the inner functions of the call controller. The fact that Mitel has made choices in selecting its numbers does not in and of itself mean that the command codes are 'original' or that they are an expression."[440] The court distinguished the Tenth Circuit's holding in Autoskill, which held copyrightable a user interface in which the user responded to the audiovisual display generated by the program by pressing the 1, 2, or 3 keys, by noting that the computer program in the Autoskill case required constant interaction between the end-user and the computer program. By contrast, in the instant case, the command codes were not part of the user interface, because the end-user did not use them to interact with a computer program. Rather, they were used only by a technician to "set up" the call controller so that the end-user could access the functions in the computer program.

In addition, the court held that the command codes were an uncopyrightable scenes a faire because "the Mitel command codes have become a common practice in the industry. Contrary to plaintiff in Autoskill, Iqtel established that it copied the command codes for reasons of efficiency."[441] The court also noted that the Mitel commands had become an industry standard, which justified treating them as an externality in permitting their copying: "I further conclude that the telephone key-pad and communications link between the long-distance carrier and the end-user impose certain parameters on the call controller market which justifies Iqtel's copying of Mitel's command codes."[442]

Finally, the court concluded that even were the command codes copyrightable, Iqtel's use of them constituted a fair use. The court did not apply each of the four fair use factors of section 107 of the copyright statute individually. Instead, citing the Ninth Circuit's decision in Sega Enterprises Ltd. v. Accolade, Inc.,[443] the court simply concluded that, although Iqtel had copied the command codes for a commercial purpose, it had a "legitimate non-exploitative purpose" for such copying in order to be able to compete with Mitel, which controlled between 75% and 90% of the call controller market.[444] Because of Mitel's dominance, the court noted that technicians who installed and programmed call controllers were accustomed to Mitel codes and were unwilling to learn new programming codes. The court also noted that the long-distance carriers were reluctant to invest in training their technicians to develop new macros and to modify their billing systems to accommodate a new call controller system. Accordingly, the court concluded that application of "the fair use doctrine's equitable rule of reason leads me to conclude that Iqtel's use of the command codes is a fair use."[445]

The Mitel case represents a significant opinion with respect to the issue of whether computer instruction sets are copyrightable. Although the case does not hold that all instruction sets are per se uncopyrightable, the court's logic that the command codes constituted an

---

[440] Mitel, 896 F. Supp. at 1055.

[441] Id. at 1056.

[442] Id.

[443] 977 F.2d 1510 (9th Cir. 1992).

[444] 896 F. Supp. 1050, 1057 (D. Colo. 1995).

[445] Id.

uncopyrightable procedure, system and method of operation because they controlled access to the functions of the controller might apply to most instruction sets. Because the court distinguished the Autoskill case on the basis that the Mitel command codes did not form part of the user interface used directly by the end-user, one might argue that the court's finding of uncopyrightability would not necessarily apply to commands, such as those involved in the Lotus v. Borland case, that form part of an end-user interface. However, the court's fair use ruling and its invocation of the scenes a faire doctrine would seem to apply to even those instruction sets that do form part of an end-user interface, at least where such instructions or commands have become a standard in the industry and must be copied for macro compatibility.

2. The Tenth Circuit's Decision

On appeal, the Tenth Circuit affirmed the district court's denial of the preliminary injunction. The Tenth Circuit noted that the district court had found Mitel's command codes unprotectable on two bases: (i) under an application of the literal language of § 102(b) of the copyright statute, on the ground that Mitel's command codes were a method of operation for matching the call controller's functions, the long distance carrier's technical demands, and the telephone customer's choices; and (ii) under the abstraction/filtration/comparison analysis.

With respect to the first base, the Tenth Circuit noted that the First Circuit had reached a similar conclusion in the Lotus v. Borland[446] case by application of the literal language of § 102(b). The Tenth Circuit characterized the First Circuit's opinion in that case as holding that "otherwise protectable expression that is embodied in a method of operation is excluded under section 102(b) from copyright protection because it is part of the method of operation."[447] The Tenth Circuit, however, rejected this broad holding by the Borland court:

> We conclude that although an element of a work may be characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection. Section 102(b) does not extinguish the protection accorded a particular expression of an idea merely because that expression is embodied in a method of operation at a higher level of abstraction. … Thus, we decline to adopt the Lotus court's approach to section 102(b), and continue to adhere to our abstraction-filtration-comparison approach.[448]

The Tenth Circuit noted, however, that in cases such as the instant one, in which the alleged infringement constitutes the admitted literal copying of a discrete, easily-conceptualized portion of a work, the court need not apply the entire abstraction/filtration/comparison analysis in detail, but rather need only determine whether the portion copied constitutes protectable

---

[446] Lotus Development Corp. v. Borland Int'l, 49 F.3d 807 (1st Cir. 1995), aff'd without opinion by an equally divided court, 116 S. Ct. 804 (1996).

[447] Mitel, Inc. v. Iqtel, Inc., 124 F.3d 1366, 1372 (10th Cir. 1997).

[448] Id. at 1372.

expression.[449] The court concluded that the command codes were not copyrightable on two grounds: originality and scenes a faire.

With respect to originality, the court concluded that Mitel used such minimal effort and judgment to select the "registers" and "descriptions" that they were unoriginal under § 102(a). The random and arbitrary use of numbers in the public domain does not evince sufficient originality to support a copyright. "Further, purely sequential elements of the codes are not original under section 102(a). The 'descriptions' are strictly sequential and matched with increasing incremental 'values.' The concept of numbering registers and descriptions in ascending sequence is analogous to arranging telephone entries in alphabetical order. See Feist, 499 U.S. at 362-63."[450] However, the court found that the efforts required of Mitel's employees to devise appropriate "values" for the wide variety of individual functions of the call controller reflected sufficient minimal creativity to qualify as an original work of authorship.[451]

Nevertheless, the Tenth Circuit held that the command codes, including the values, should not be protected under the scenes a faire doctrine. The Tenth Circuit first noted that the district court had partially erred in its analysis of the scenes a faire doctrine by focusing on "whether external factors such as market forces and efficiency consideration justified Iqtel's copying of the command codes. The court's analytical focus should have remained upon the external factors that dictated Mitel's selection of registers, descriptions, and values."[452] However, the Tenth Circuit went on to rule that the district court had correctly found that much of the expression in Mitel's command codes was dictated by the proclivities of technicians and limited by significant hardware, compatibility, and industry requirements.

> External factors frequently dictated Mitel's selection of particular values to activate the range of call controller functions. For example, many of the values were selected by Mitel's product management department in response to customer demand or to ensure compatibility with equipment already installed in the central offices of Mitel's customers. Frequently, the values were divided in equal increments across a numerical range, and the descriptions and the value increments were matched in ascending steps. Standard programming conventions such as '1' for 'on' and '0' for 'off' determined some of the descriptions and values. In addition, some of the values for the set of command codes that were actually copies were dictated by the need for compatibility with older-model Mitel call controllers or the limits on the capabilities of the controller itself. Other values were dictated by the limits inherent in the public telephone networks that the call controllers accessed.[453]

---

[449] Id. at 1373.

[450] Id. at 1374.

[451] Id.

[452] Id. at 1375 (emphasis in original).

[453] Id. (citations omitted).

Accordingly, the Tenth Circuit concluded that the Mitel values, although non-arbitrary original expression, were unprotectable as scenes a faire because they were dictated by external functionality and compatibility requirements of the computer and telecommunications industries.[454] Because of its ruling under the scenes a faire doctrine, the court concluded that it need not review the district court's ruling that Iqtel's copying of the Mitel command codes constituted a fair use.

The Tenth Circuit's focus under the scenes a faire doctrine on the original creator's perspective, rather than the second comer, in deciding whether the expression at issue was an industry standard or dictated by an externality, causes the scenes a faire doctrine to have a narrower applicability to justify copying expression that has become an industry standard since it was first created. Under the Tenth Circuit's approach, a factual inquiry must be made concerning whether the original creator of the expression at issue was constrained by externalities. The Tenth Circuit found that Mitel was so constrained in this particular case, but had it not been, the implication is that even the arbitrary or random selection of the command code values would have been sufficient creativity to be protectable. Thus, the Tenth Circuit's interpretation of the scenes a faire doctrine – unlike that of the district court – is not sufficiently broad to imply that all sets of command codes or instruction sets that are copied for compatibility are necessarily unprotectable.

K. THE CONTROL DATA CASE

A 1995 case raised the question whether an "emulator" program infringed the copyright in the program it was designed to emulate. In Control Data Sys. v. Infoware, Inc.,[455] Control Data owned the copyright in a "Network Operating System" (NOS) for use on Control Data's "Cyber" mainframe computers. The defendant introduced a product known as "AlphaCyber," which was designed to be an "emulator" of NOS that would permit customers to use application programs designed for NOS on hardware other than the Cyber computer line. Control Data sought a preliminary injunction against the use of AlphaCyber.

Following the Tenth Circuit's analytical approach in Gates Rubber, the court noted that the plaintiff had to establish (1) whether the defendant, as a factual matter, copied portions of the plaintiff's program, and (2) whether, as a mixed issue of fact and law, those elements of the program that have been copied are protected expression and of such importance to the copied work that the appropriation is actionable.[456] Control Data alleged copying by the defendant of several elements of NOS: literal copying of lines of the NOS source code; copying of the NOS input and output formats; copying of the NOS file layouts; copying of NOS source code parameters; and copying of NOS commands.

Applying the abstraction/filtration/comparison test of Gates Rubber, the court turned to whether the allegedly copied elements of NOS constituted protectable expression or should be

---

[454] Id. at 1376.

[455] 903 F. Supp. 1316 (D. Minn. 1995).

[456] Id. at 1320.

filtered out. The defendant argued that the similarities that existed between its program and NOS were "dictated by the fact that the purpose of AlphaCyber, or its 'idea,' is to function just like NOS", and argued that the elements necessary to the idea of making a product function like NOS were "irretrievably merged into that idea and must be filtered from the analysis."[457]

Control Data challenged the defendant's view of the merger doctrine, arguing that the proper approach was to "compare the idea of NOS, that is, the idea of making an operating system for the Cyber computer, with the expression of that idea as embodied in NOS. Because there are many different ways in which the allegedly copied elements of NOS could have been expressed, it asserts, those elements are protectable, and the merger doctrine does not apply."[458] Without any additional analysis, the court simply stated that it found Control Data's argument to be "persuasive."[459] Similarly, the court rejected the defendant's reliance on the "externality doctrine," ruling that the "question to be examined is whether external factors limited the choices available to NOS programmers, not whether external factors may somehow limit the choices of AlphaCyber programmers. As Control Data points out, Infoware fails to list any external factors that limited the freedom of expression of the programmers who wrote NOS."[460] Thus, the court was unwilling to view NOS itself as an "externality" to which the defendants had to conform their own competing program.

With respect to the comparison step, the court found that there was a substantial likelihood, based on the evidence submitted, that Control Data would be able to demonstrate that the protected portions of NOS that the court found were copied were qualitatively significant to NOS.[461] Accordingly, Control Data had established a likelihood of success on its infringement claim, and the court issued a preliminary injunction in favor of Control Data.

Because much of the evidence submitted in this case was subject to a protective order, there was very little factual discussion set forth in the court's opinion to support the court's conclusions. It is therefore difficult to judge the potential reach of this decision with respect to other "emulator" products. Control Data had alleged literal copying of over 2000 lines of the source code of NOS,[462] and this fact alone may have justified a preliminary injunction, if such lines were not unprotectable because dictated by efficiency, subject to the merger doctrine, or subject to other limiting doctrines. However, the other allegedly copied elements – NOS input/output formats, NOS file layouts, NOS source code parameters, and NOS commands – may have been necessary for emulation compatibility. The court's opinion does not say, or give any separate analysis of, whether each of these elements was protectable by Control Data's copyright. Accordingly, it is unclear what implication this decision may have for cases in which there is no literal copying of source code, but other elements of a program (such as input/output

---

457 Id. at 1323.

458 Id.

459 Id.

460 Id.

461 Id. at 1324.

462 See id.

formats, commands, and file formats) may have been copied in order to achieve emulation compatibility.

L. THE BATEMAN CASE

A very interesting case decided in 1995 by the Eleventh Circuit, Bateman v. Mnemonics, Inc.[463] was the first to explicitly consider which jury instructions should be given under the abstraction/filtration/comparison test. As previously discussed in connection with the Apple v. Microsoft and Capcom v. Data East cases, a central question left open by the look and feel cases to date is whether, having identified unprotectable elements in a copyrighted computer program, those elements should be presented to the jury in the adjudication of substantial similarity and, if so, how the jury should be instructed to treat those unprotectable elements in making its similarity comparison.

The Bateman case does not resolve this issue, for it is unclear from the Eleventh Circuit's opinion what the jury was allowed to see. The jury was in effect instructed, however, to ignore unprotectable elements in its adjudication of substantial similarity and to compare only "protectable expression" in the copyrighted work to the defendant's work.[464] But the instruction at issue in the appeal did not instruct the jury what constituted "protectable expression" – in particular, whether such expression could be comprised of combinations at higher levels of abstraction of elements ruled unprotectable at lower levels of abstraction. Accordingly, the Bateman decision leaves many questions unanswered with respect to how properly to instruct a jury with respect to handling unprotectable elements in a look and feel case.

The Bateman case also raised the very important compatibility issue of whether the technical interface between an operating system and applications written to run under that operating system is protectable by copyright. The Bateman court did not issue a definitive ruling on this issue, however. The court stated only that "interface specifications" are not, as a matter of law, uncopyrightable, although it noted that compatibility requirements may negate a finding of infringement for alleged copying.[465]

1. Factual Background

The plaintiff was the developer of a single board computer operating system (SBCOS) that ran on a single board computer used in automated parking systems. Several of the boards containing SBCOS were sold by one of the defendants, BCS, Inc., under a license from the plaintiff to one of the other defendants, Generex Corporation, which developed an application program designed to interoperate with SBCOS using specifications that described the system calls necessary to communicate with the operating system. Generex Corporation was subsequently acquired by another of the defendants, Parking Automation Corporation (PAC).

---

[463] 79 F.3d 1532 (11th Cir. 1995).

[464] Id. at 1544.

[465] Id. at 1547.

The plaintiff terminated BCS's license to distribute copies of SBCOS. Thereafter, PAC requested a copy of the source code for SBCOS so that PAC could develop a compatible operating system in order to be able to continue to execute the application program that Generex had developed. When the plaintiff denied PAC's request for source code, PAC disassembled SBCOS and identified those elements of SBCOS that were necessary for compatibility with its application program, then proceeded to develop a new operating system, which PAC called the Lane Control Computer Operating System (LCCOS), that would run its application program. PAC admitted that it copied portions of the SBCOS code during the design and development of LCCOS,[466] although it is not clear what these portions were, and whether they were copied only for intermediate purposes, or whether they ended up in the final version of LCCOS. Although the court does not explicitly state so, it is apparent that LCCOS had many, or perhaps all, of the same system calls as SBCOS.

The plaintiff filed suit for copyright infringement against various defendants, alleging that LCCOS infringed the plaintiff's copyright in SBCOS. The jury found infringement and awarded damages. The defendants appealed, challenging the district court's instructions to the jury under the abstraction/filtration/comparison test, and arguing that the SBCOS "interface specifications" were not copyrightable. Although the parties and the court used the term "interface specifications," the Eleventh Circuit's opinion does not define what was meant by that term. Presumably, however, in view of the compatibility issues that were at stake in the case, the term was not used by the court to refer to merely a hardcopy printout of a technical specification for SBCOS. Rather, the court seems to be referring to the technical interface – and most likely the system calls – to SBCOS, which LCCOS would have had to copy in whole or in part to achieve compatibility with the application program that was originally written to run under SBCOS.

2. The Jury Instruction on Filtration

Before Bateman, the Eleventh Circuit had not spoken on whether it would adopt the abstraction/filtration/comparison analysis of the Altai case, although two district court decisions in the Eleventh Circuit's jurisdiction (the CMAX case and the Mitek case discussed below) had previously adopted such analysis. In Bateman, the Eleventh Circuit did not state that it was explicitly adopting the abstraction/filtration/comparison analysis of Altai for use as a methodology in software copyright infringement cases. However, the district court's jury instructions were based on the Altai analysis, and the Eleventh Circuit accepted the district court's general approach embodied in the jury instructions without challenge in considering the correctness of the instructions.[467]

---

[466] Id. at 1542.

[467] The Eleventh Circuit stated: "Throughout their briefs, the parties refer to this test as the Nimmer 'successive filtration' test. In instructing the jury, the district court also referred to this test as the 'Nimmer test.' ... While we certainly recognize that the Altai court endorsed Professor Nimmer's 'successive filtering method' for separating protectable expression from nonprotectable material, the test set forth in Altai is more comprehensive than that formulated by Professor Nimmer. The jury instructions were clearly based on the Altai

The district court issued the following instruction (in relevant part) to the jury with respect to the "filtration" step of the analysis of infringement:

> Under the Nimmer test, substantial similarity of the *non-literal elements* is determined by comparing with the defendants' program, that protectable expression of the copyrighted work which remains after filtering out any portion of the copyrighted work, which represents only ideas, elements[] dictated solely by logic and efficiency, elements dictated by hardware or software standards, computer industry programming and practices or elements which are taken from the public domain.[468]

The defendants challenged this instruction on appeal on the ground that it was error to instruct the jury to filter out only nonliteral similarities in applying the abstraction/filtration/comparison analysis. As is evident from the italicized language above, the jury instruction referenced substantial similarity of "non-literal elements" only, which the defendants contended, and the Eleventh Circuit agreed, misled the jury to believe that any instances of literal copying of the plaintiff's code by PAC were by definition acts of copyright infringement. The Eleventh Circuit emphasized that all unprotectable material must be filtered out based on the limiting doctrines of "challenges to originality, such as merger, scenes a faire, standard technique or practice, considerations of efficiency, compatibility requirements, and the like."[469]

3. The Absence of a Jury Instruction on Compatibility

The defendants also alleged on appeal that the district court erred in not instructing the jury on the legal consequences of a finding that certain instances of literal copying of code by PAC were dictated by compatibility and interoperability requirements. Specifically, the Eleventh Circuit noted that the "district court gave no instruction on whether the application

---

version of the Nimmer test, and therefore any reference in this opinion to the 'Nimmer test' is to be understood as a reference to the Altai version of the Nimmer test." Id. at 1543 n.24.

[468] Id. at 1544 (emphasis added).

[469] Id. at 1545. "We express no opinion as to whether it is better to consider these challenges to literal copying as part of the filtration step or rather to consider them in a separate, yet parallel analysis. The important point here is that such an analysis is necessary – it makes little difference which methodology is employed." Id.

It is interesting that the Eleventh Circuit characterized all of the enumerated limiting doctrines as "challenges to originality." The Eleventh Circuit viewed all such limiting doctrines as emanating from the fundamental constitutional requirement of originality for copyright protection, as articulated by the Supreme Court in Feist Publications v. Rural Tel. Serv. Co., 499 U.S. 340 (1991). Specifically, the Eleventh Circuit cited the passage in Feist in which the Supreme Court stated that to establish a claim of copyright infringement, the plaintiff must prove ownership of a valid copyright and "copying of constituent elements of the work that are original." Bateman, 79 F.3d at 1541 (quoting Feist, 499 U.S. at 361).

program/SBCOS interface at issue in this case was a constraint that rendered the interface either unprotectable or subject to a fair use analysis."[470]

PAC had proposed a jury instruction stating that "you are to filter out as unprotectable expression portions of the program that the evidence shows were dictated by the interface with the applications code which was an external constraint."[471] The district court refused this instruction, but did offer the following instruction: "[C]omputer programs are, in essence, utilitarian articles. Articles that accomplish tasks. As such, they contain many structural and visual display elements that may be dictated by the function to be performed, by considerations of efficiency, or by external factors, such as compatibility requirements and industry demands."[472]

On appeal, the defendants challenged the district court's refusal to issue their requested instruction. In addition, the defendants argued on appeal that the plaintiff's "interface commands" were uncopyrightable and, thus, PAC was not obligated to avoid copying them by rewriting its application program.[473] The opinion does not state what the term "interface commands" meant, or how that term was different from "interface specification," but presumably both terms were meant to refer to the system calls of SBCOS.

In response, the Eleventh Circuit ruled that, although the district court's jury instruction was "not technically incorrect," it failed "to instruct the jury on the legal consequences of a finding that certain copying of code by PAC was dictated by compatibility requirements."[474] The Eleventh Circuit further stated:

> The Sega and Altai courts are certainly not alone in noting that external factors such as compatibility may work to deny copyright protection to certain portions of a computer program. ... Whether the protection is unavailable because these factors render the expression unoriginal, nonexpressive per 17 U.S.C. §102(b), or whether these factors compel a finding of fair use, copyright estoppel, or misuse, the result is to deny copyright protection to portions of the computer program. Thus, we today join these other circuits in finding that external considerations such as compatibility may negate a finding of infringement.[475]

The Eleventh Circuit rejected, however, the defendant's argument that interface commands are per se uncopyrightable. "It is an incorrect statement of the law that interface specifications are not copyrightable as a matter of law. ... We need not decide whether PAC is correct in its assertion that, given the particular facts of this case, it was not obligated to rewrite

---

[470] 79 F.3d at 1546.

[471] Id.

[472] Id.

[473] Id. at 1547.

[474] Id. at 1546.

[475] Id. at 1547.

its application program to avoid copying Bateman's interface specifications. PAC, however, is incorrect in arguing that this rewriting was not required because Bateman's interface specifications are not entitled to copyright protection as a matter of law."[476]

Thus, the Eleventh Circuit was unwilling to rule that operating system calls constitute uncopyrightable subject matter. It was, however, willing to recognize that under particular facts, the copying of such system calls might not constitute infringement under any of several doctrines – originality, section 102(b) of the copyright statute, fair use, copyright estoppel, or misuse. The court explicitly left open any more detailed discussion about the circumstances under which such copying might not constitute infringement, and remanded to the district court for a new trial.[477]

M. THE COMPAQ CASE

The case of Compaq Computer Corp. v. Procom Technology, Inc.[478] addressed the question whether certain parameters and threshold values used by a computer program to monitor when a hard drive is about to fail are protectable by copyright. This case is one of the few to date to raise issues about copying of elements of a computer program for reasons related to *hardware* compatibility.

1. Factual Background of the Case

The plaintiff Compaq Computer Corp. developed a computer program known as the "Compaq Insight Manager" (CIM) to monitor the performance of the hard drives running in Compaq's "ProLiant" brand of servers. CIM had the ability to generate "prefailure warnings" to the system administrator – indications that a drive has reached a point in its life where failure may be imminent. The prefailure warnings were generated when five particular monitored parameters for the hard drive fell below certain predefined threshold values. The threshold values used by the CIM program were stored in a portion of the hard drive known as the "Monitor and Performance Partition" (M&P Partition).

Compaq registered the threshold values – along with other data stored in the M&P Partition – for three of its hard disk drives with the Copyright Office. Because the data was submitted in machine readable format, the Copyright Office issued the registration under its "Rule of Doubt," the legal effect of which was to deny Compaq the presumption of validity of the copyright normally accorded by a copyright registration.

---

[476] Id. at 1547 & n.31.

[477] The court stated that a finding of noninfringement "will depend on the particular facts of a case, and thus it would be unwise for us to try to formulate a bright-line rule to address this issue, given the importance of the factual nuances of each case. In no case, however, should copyright protection be extended to functional results obtained when program instructions are executed and such results are processes of the type better left to patent and trade secret protection." Id. at 1547 n.33.

[478] 908 F. Supp. 1409 (S.D. Tex. 1995).

Compaq's servers were able to function both with hard drives manufactured by Compaq and with hard drives manufactured by third parties. Compaq intended for all its disk drives to be sold with a copy of the threshold values stored in the M&P Partition. The prefailure warning system of the CIM program would not function without the threshold values stored on the disk drive. When a new drive is added to a ProLiant server, the network administrator runs a Compaq program called EISA Config. EISA Config determines whether the drive is a Compaq drive by checking the vendor ID stored in the firmware.[479] If so, and if the drive does not contain the threshold values, EISA Config stores those values on the drive.

The defendant Procom distributed a line of hard drives designed for use with Compaq's ProLiant servers. Because Procom's customers valued the features offered by CIM, Procom set out to discover how it could modify its drives to be compatible with CIM. Procom discovered that if it modified the vendor ID of its drives to identify them as Compaq drives, EISA Config would then store the threshold values in the drive. Procom decided, however, that it was not comfortable marketing a drive that incorrectly identified the vendor of the drive as Compaq. Instead, Procom was able to determine that the M&P Partition contained data which enabled the prefailure warnings. Although Procom did not understand what the data in the M&P Partition represented, it was able to copy the necessary data to its drives and thereby enable the warnings.

Compaq sued Procom for copyright infringement. After the suit was filed, Procom stopped copying the threshold values onto its hard drives, and instead decided to change the vendor ID of its drives to "Compaq" to enable the threshold values to be written by the system administrator using the EISA Config program. This change had the effect, however, of causing a portion of the screen display generated when CIM was run to erroneously identify the vendor of the drive as Compaq.

2. Copyrightability of the Threshold Values

Procom asserted in the lawsuit that the threshold values are not copyrightable expression or, alternatively, that Procom's use of the threshold values was permissible under the doctrines of merger, fair use, and scenes a faire. The district court rejected these arguments. With respect to the question whether the threshold values constituted copyrightable expression, the court held that Compaq's compilation of the five threshold values did constitute copyrightable expression under the Supreme Court's decision in Feist,[480] because Compaq had exercised discretion in choosing both the number of parameters to monitor – five – and which five particular parameters those would be.

The court rejected Procom's argument that Compaq's compilation could not be copyrighted because, although Compaq may have had options in selecting the parameters to track, once those parameters were identified, competitors could not choose alternatives if they wanted their drives to be fully compatible with CIM. The court ruled that this argument

---

[479] All hard drives contain a vendor ID string in the drive's firmware. The placement of the vendor ID in the firmware is dictated by an industry standard called the "Small Computer System Interface" (SCSI – pronounced "scuzzy") standard.

[480] Feist Publications v. Rural Tel. Serv., 499 U.S. 340 (1991).

incorrectly focused on the choices available to the copier, rather than the author. The issue for copyrightability is whether the author had any choices, not whether the copier did.[481]

The court also noted that Compaq's threshold values were not facts because they were not empirically verifiable. Instead, the court held that the particular threshold values selected by Compaq were the result of a decision making process by Compaq based on an estimate of when the drive would actually fail and a business judgment as to the point in the life of the drive that Compaq was willing to replace it under its warranty program.[482] Accordingly, the compilation of five threshold values was copyrightable.[483]

3. Merger

With respect to Procom's merger argument, the court noted that a third party attempting to gain access to CIM has no choice but to also select the same five parameters for observation, for if other parameters were selected, any warnings issued by CIM would be meaningless. Nevertheless, the court held there was no merger, because there "are numerous ways that a drive supplier may express its opinion as to when it should replace its drives"[484] based on monitored parameters. The court noted that Compaq's threshold values did not simply represent the point at which the drive *will* fail, but rather the point that Compaq deemed most optimal to replace the drives.

4. Fair Use

The court also rejected Procom's argument that its use of the parameters constituted a fair use. Applying the first statutory fair use factor – whether the copy was made for commercial use or for some other purpose – the court stated that "the court should move beyond a simple inquiry of commercial versus noncommercial purpose and instead consider whether the challenged use promotes the purposes of copyright law."[485] The court distinguished Procom's copying from the copying that the Ninth Circuit found to be a fair use in Sega Enterprises Ltd. v. Accolade, Inc.[486] In Sega, Accolade disassembled the code in Sega's video game cartridges to determine what initialization code was required by the Sega game console's security mechanism in order to execute a game cartridge. Accolade included the initialization code (which consisted of the word "SEGA") in its own game cartridges so that they would work with the Sega console.

---

[481] 908 F. Supp. at 1418.

[482] Under Compaq's warranty program, Compaq would replace the drive when the CIM program indicated that it was about to fail.

[483] Id.

[484] Id. at 1419.

[485] Id.

[486] 977 F.2d 1510 (9th Cir. 1992).

The court in the Compaq case emphasized that the disassembly performed by Accolade in the Sega case was intermediate copying only, done for the purpose of studying the functional requirements for compatibility with the Sega console. By contrast,

> Procom's use of Compaq's threshold values was not an intermediate use, intended to facilitate the study of functional aspects of CIM. As in Sega, Procom made a verbatim copy of the copyrighted material. However, Procom never used the copy to develop its own, noninfringing product. Instead Procom simply reproduced the copied data onto its own drives to achieve interoperability.[487]

Accordingly, the court concluded that, because the particular use made by Procom of the copyrighted material was also the ultimate use, and the ultimate use was clearly commercial, the first statutory factor weighed against Procom.

With respect to the second statutory factor – the nature of the copyrighted work – the court noted that the threshold values had both functional and expressive components. They were functional to the extent they had to be copied for access to the prefailure warning feature of CIM. They were expressive to the extent that third parties did not need to use the same five numbers that Compaq used in order to receive prefailure warnings in general (although not from *CIM*). Accordingly, the court determined that the second factor did not weigh in favor of either party.

With respect to the third statutory factor – the amount and substantiality of the copying – the court held that Procom copied "the core, or the heart, of Compaq's copyrighted material" in view of the fact that the "only section of the M&P partition which embodies any real expression is the threshold value segment and this is the portion that Procom cloned."[488] Accordingly, the third factor favored Compaq.

With respect to the fourth statutory factor – the effect of the use upon the potential market for or value of the copyrighted work – the court again distinguished the Sega case. In Sega, Accolade sought only to produce games that legitimately competed with Sega's games, the success of which would be determined by the characteristics of the game program developed by Accolade. In contrast, "once Procom successfully copies Compaq's threshold values, Procom's product is a virtual duplicate of Compaq's product. The result of Procom's copying is to have a direct and adverse effect on the market for Compaq's work."[489] Accordingly, the court held that the fourth factor weighed strongly in favor of Compaq.

In sum, the court concluded that the net of the four statutory factors weighed against a finding of fair use: "Procom made no attempt to understand how the numbers facilitated interoperability, let alone to understand the meaning of each particular number. To permit such a use as a fair use would be counter to the purpose of the Copyright Act."[490]

---

[487] 908 F. Supp. at 1420.

[488] Id. at *1421.

[489] Id.

[490] Id.

5. Scenes a Faire

The court also rejected Procom's argument that the threshold values should be treated as dictated by the "external factor" of compatibility under the scenes a faire doctrine.[491] Procom argued that because CIM dictated the order in which the threshold values had to appear on the hard drive, the *ordering* of the values could not be copyrighted. The court agreed:

> In order to obtain prefailure warnings from CIM, the drive must contain five numbers in correct place on the drive. However, the warnings must be meaningful before a drive is truly compatible with CIM. Since the warnings are based on the tracking of five specific parameters, a third party seeking compatibility with CIM has no choice but to use those five parameters as well. In addition, the numbers representing those parameters must be ordered on the drive in the specific manner that CIM expects.[492]

Accordingly, the court determined that this "method of organizing CIM is a compatibility requirement that cannot be protected by copyright law" under section 102(b) of the copyright statute.[493]

The court further held, however, that while the *ordering* of the threshold data was not protected, the *specific values* chosen by Compaq were protected. "A third party who understands the workings of CIM and hard drives could potentially develop its own threshold values. While duplication of Compaq's numbers will produce the same results as would a Compaq drive, this is not required for interoperability and thus, the numbers themselves are protectable."[494] Accordingly, the court ruled that Procom had infringed Compaq's copyrights by copying the Compaq threshold values onto its hard drives.

The practical effect of the holding of the court that the specific values of Compaq's threshold data were protectable was to require Procom to understand the meaning of the parameters Compaq had chosen to monitor – which it would presumably have to do through disassembly and other reverse engineering – and then to choose a set of its own five specific values for those parameters on which CIM would trigger a prefailure warning. Curiously,

---

[491] "Elements of a program that have been dictated by external factors are also denied protection under [the scenes a faire] doctrine. ... In the context of computer programs, these external factors include such considerations as 'hardware standards and mechanical specifications, software standards and compatibility requirements....'" Id. at 1421 (citations to the Tenth Circuit's decision in Gates Rubber omitted).

[492] 908 F. Supp. at 1421.

[493] Id. at 1422. "Moreover, the use of five numbers to access CIM is probably best characterized as a system. Under § 102(b) of the Copyright Act, systems, methods, and procedures are not copyrightable – only their expression is, and only to the extent that there are several means of expression." Id. at 1419 n.12.

[494] Id. at 1422.

however, this effect seems inconsistent with what the court ruled in rejecting Compaq's argument that Procom's use of "Compaq" as the vendor ID in its drives in order to circumvent the need to copy Compaq's copyrighted threshold values was a trademark infringement. The court rejected Compaq's trademark argument because the court found such use of Compaq's vendor ID to be a functional use. The court supported its ruling of functionality in part on the basis of the following factual finding: "While it is certainly conceivable that Procom could develop its own threshold data, there is no evidence in the record to show that this is commercially feasible. Accordingly, the use of Compaq's trademark has the functional aspect of locking out users from access to CIM."[495] If it was not commercially feasible for Procom to develop its own threshold data, then it seems curious why this fact did not affect the court's fair use analysis.

6. Contributory Infringement

Compaq also asserted a contributory copyright infringement claim against Procom based on Procom's modification of the vendor ID in its drives to read "Compaq." When such drives were first used in a ProLiant server, EISA Config would identify the drive as a Compaq drive and copy Compaq's copyrighted threshold values onto the drive. Compaq contended that Procom's sale of these drives constituted a contributory infringement of Compaq's copyrights.

Procom argued in defense that there was no contributory infringement because its drives had substantial non-infringing uses. The court noted that Procom's argument focused on the wrong issue, which was not whether there were substantial non-infringing uses of the entire drive, but rather whether there were substantial non-infringing uses of the vendor ID feature:

> Procom's sole use for modifying the vendor ID portion to read Compaq is to cause the threshold values to be written to Procom drive. Since Procom has shown no substantial non-infringing use for the modification of its drives to identify Compaq as the vendor, the Court finds that Procom's actions constitute contributory infringement.[496]

The court also rejected Procom's argument that a statement in the license agreement between Compaq and purchasers of ProLiant servers, acknowledging that some programs might not run as effectively or might cause errors in data or operations when the programs were used on non-Compaq products, implicitly permitted the copying of the threshold values onto third party drives. "This statement merely acknowledges that while the ProLiant server is compatible with third party products, not all of those products will run as effectively with the ProLiant as

---

[495] Id. at 1423.

[496] Id. at 1424. Procom also argued that the copying of the threshold data onto the Procom drives was authorized under § 117(1) of the copyright statute. The court rejected this argument, holding that § 117(1) was inapplicable to the facts of the case since purchasers of Procom drives were not copying or adapting EISA Config or any other program in order to utilize that program with a machine. Id.

would Compaq products. This statement does not support the conclusion that Compaq intended to allow copying of the threshold values onto third party drives."[497]

The court issued an injunction enjoining Procom from copying or distributing unauthorized copies of Compaq's copyrighted data compilations and threshold values and from manufacturing or offering for sale hard disk drives which cause the reproduction of such copyrighted material.

7. Analysis

The court's rulings in the Compaq case apply several of the limiting doctrines of copyright more narrowly than did the court in the Mitel case, which presented similar issues. In Mitel, the court was willing to consider the Mitel command codes to be uncopyrightable scenes a faire because those command codes had become an industry standard, which justified treating them as an externality in permitting their copying, even though a competing manufacturer of call controllers could certainly have achieved the same functionality using a differently defined command set. Yet, even though Procom similarly argued that Compaq's threshold values were part of the industry standard established by the CIM program, the court was willing to hold only that the *ordering* of the threshold values constituted scenes a faire, but not the values themselves.

Similarly, the Mitel court was willing to apply the fair use doctrine to the copying of the Mitel command codes since such copying was found to have been performed in order to compete with Mitel, which controlled between 75% and 90% of the call controller market. The court seemed heavily influenced by the fact that Iqtel's customers, who were experienced at programming Mitel controllers, demanded that Iqtel controllers be command compatible. The Compaq court was, however, unwilling to accord similar weight in applying the fair use doctrine to customers' demands for CIM compatibility. It seems likely that customers using Compaq-compatible hard drives would want CIM to issue prefailure warnings that would operate the same as they would in the case of a Compaq drive. Indeed, as previously noted, the Compaq court stated that there had been no evidence that it was even commercially feasible for Procom to develop different threshold values. Nevertheless, the Compaq court required Procom to have different thresholds for the prefailure warnings in order to avoid copyright infringement.

The Compaq and Mitel cases illustrate a fundamental issue that recurs in the "copying for compatibility" cases – namely, in determining whether a limiting doctrine should apply to permit copying of an element of a copyrighted work that has become a standard, should one focus on the choices available to the original developer of the copyrighted work, or to the choices available to the copier who wishes to achieve compatibility? If one focuses on the choices available to the original developer, then one might easily reach the conclusion that the now-standard element contained in the original developer's copyrighted work constitutes protectable expression, because the original developer exercised choice in creating that element in the first place. This is how the Compaq court concluded that the threshold values were protectable. Judge Keeton in the Borland lower court decisions used similar reasoning to conclude that the Lotus 1-2-3 commands were copyrightable, for Lotus had exercised creativity and choice in

[497] Id. at 1425.

naming and ordering the commands of the Lotus 1-2-3 menu hierarchy. Similarly, the district court in the Control Data case, in ruling on the question whether a network operating system interface was copyrightable, noted that the issue was whether external factors limited the choices available to the programmers who created the original operating system, not whether external factors somehow limited the choices of the programmers seeking to create a compatible operating system.

Under this reasoning, copying can usually be excused only if some other limiting doctrine applies, such as the fair use doctrine. Courts following this approach also tend to define the "idea" underlying the copyrighted work at higher levels of abstraction, and view the element that must be copied for compatibility as separable "expression" of the more abstract idea, by virtue of the choices available to the developer of such element.

By contrast, if one focuses on the choices open to the later developer who wishes to achieve compatibility with the original standard, then one might reach the conclusion that such later developer has no choice but to copy certain preexisting elements that are necessary to achieve compatibility. Courts focusing on the issue from the perspective of the later developer of a compatible product have been more willing to view the elements necessary for compatibility as an uncopyrightable scenes a faire, or an uncopyrightable system or method of operation. The First Circuit in the Borland case, as well as the Mitel court, adopted this approach. Courts that follow this approach also tend to define the "idea" underlying the copyrighted work at lower levels of abstraction, and view the element that must be copied for compatibility as an inseparable part of that "idea." Under this approach, the fair use doctrine is not needed to excuse the copying, although some courts (such as the Mitel court) have also applied the fair use doctrine as an alternative ground for permitting the copying.

As the recent cases illustrate, this split in fundamental approach to defining an "idea" and applying the limiting doctrines can lead to widely varying results as to what may be copied for compatibility purposes. The Borland case afforded an excellent opportunity for the Supreme Court to lend some guidance in this area. This guidance was unfortunately not forthcoming in view of the equally divided Court.

N. THE MITEK CASE

A 1996 case from the Eleventh Circuit addressed the scope of copyright protection for various nonliteral elements of a computer program, most of which related to functional *features* of the program. As in the earlier Bateman case (discussed above), the Eleventh Circuit did not state that it was explicitly adopting the abstraction/filtration/comparison analysis of Altai for use as a methodology in software copyright infringement cases. However, the district court applied the Altai analysis, and the Eleventh Circuit reviewed the district court's application of that analysis to the facts of the case without questioning the appropriateness of the analysis.[498]

ANALYSIS OF THE DISTRICT COURT'S DECISION

[498] In footnote 14 of the Court's opinion, the Eleventh Circuit cited the Bateman case with a parenthetical characterizing that case as "discussing and applying the Altai test".

Drawing heavily on the Engineering Dynamics and Gates Rubber cases, the district court in the 1994 decision of MiTek Holdings, Inc. v. Arce Engineering Co., Inc.[499] adopted the abstraction/filtration/comparison approach of the Altai case, and adopted a "virtual identicality" standard for comparison of the works as a whole, following the Apple v. Microsoft cases. After conducting a bench trial, the court ruled that five nonliteral elements of the plaintiff's computer program were protectable and were similar to elements in the defendant's program. However, the court ultimately found no infringement because the similarities with respect to such five elements were a *de minimis* part of the plaintiff's program.

1. Factual Background

The plaintiff was the owner of the copyright in several versions of a layout program called "Aces" designed to draw an architectural blueprint, indicating the size and location of wood trusses on the walls of a structure. All versions of the Aces program were written to run under the MS-DOS operating system. The Aces program was written by an employee named Sotolongo for a company ultimately acquired by the plaintiff MiTek Holdings, and was first published in 1989. In 1991, Sotolongo was hired by the defendant Arce Engineering Co. and told to develop "from scratch" a wood truss layout program called "Arce" that would run under the Windows operating system.

The plaintiff filed a copyright infringement suit against the defendant, alleging 18 similarities between the programs, many of which related to functional *features* of the programs, but which also included alleged similarities in menu command structures and various visual elements of the user interface.

2. Adoption of the Abstraction/Filtration/Comparison Test

The district court noted that the Eleventh Circuit had not yet directly addressed the copyrightability of nonliteral elements of computer programs. Stating its belief, however, that copyright protection should protect both the code of the program and "the way the program looks, sounds and interacts with the user,"[500] the court proceeded to adopt for analysis of infringement the "approach that appears to have gained the widest acceptance" – the abstraction/filtration/comparison test "first articulated by the Second Circuit in Altai and most recently adopted by the Fifth Circuit in Engineering Dynamics and by the Tenth Circuit in Gates Rubber."[501]

The court largely adopted the formulation of the abstraction/filtration/comparison test articulated in the Gates Rubber case. With respect to the filtration step, the court noted that the following must be filtered out as unprotectable to "obtain a core of protectable expression":[502]

---

[499] 864 F. Supp. 1568 (S.D. Fla. 1994).

[500] Id. at 1577.

[501] Id.

[502] Id. at 1578.

ideas, processes, methods or scientific discoveries, facts, information in the public domain, and scenes a faire – i.e., "expressions that are common to a particular subject matter or are dictated by external factors."[503]

The court added a fourth step to the abstraction/filtration/comparison test, an application of the *de minimis* doctrine: "[I]f the Court does find substantial similarity between certain elements of the programs, the Court will determine if the Defendant misappropriated a substantial portion of Plaintiff's programs."[504]

Citing one of Judge Walker's decisions in the Apple v. Microsoft case and the Ninth Circuit's Harper House decision, the court also noted that where a work consists "largely of uncopyrightable elements," a standard of "virtual identicality" or "bodily appropriation" must be used in the comparison step to adjudicate infringement.[505]

3. Application of the Test

The court determined that it need not undertake the abstraction portion of the test because the plaintiff had identified 18 specific nonliteral elements of its program that it contended were copied and were entitled to copyright protection. Accordingly, the court confined its analysis to a determination whether each of such 18 elements were protectable or not under the filtration step. The following table summarizes the court's rulings with respect to the 18 allegedly similar nonliteral elements:

---

[503] Id.

[504] Id. at 1579.

[505] Id. at 1584.

TABLE IV – COPYRIGHTABILITY OF 18 ALLEGED SIMILARITIES

SIMILARITY	PROTECT-ABLE?	LIMITING DOCTRINE	THE COURT'S RULING
1. The main menu command system, including the main menu and submenu command tree structure. 2. The submenu command and function organization.	No	Process dictated by externality	With respect to structure, the court ruled that the "method the Aces Layout Programs follow, including the menu and the sub-menu command tree structure, is a process that is not entitled to copyright protection" because such method "mimic[s] the steps a draftsman would follow in designing a roof truss plan by hand."[506] With respect to visual similarities, the court found no substantial similarity because the plaintiff's program used words and abbreviations for commands, and the defendant's program depicted commands by icons.[507]
3. The fact that the program was developed for IBM and compatible personal computers. Existing layout programs had been developed for other, more expensive computers.	No	Idea	The court ruled that choosing a particular system compatibility is relevant only to the literal aspects of the program, which were not alleged to have been copied.[508]

[506] Id. at 1580.

[507] Id.

[508] Id. at 1581.

4. "Dynamic sizing": the program's ability to automatically fit the drawing into the space available on the screen, using as much of the screen as possible.	No	"Public domain" (lack of originality)[509]	Because many other graphics programs had this capability, "the Court finds that this feature is part of the public domain and is therefore not entitled to copyright protection.[510]
5. The "pop-up" 15-key number pad that appeared in the drawing area of the screen whenever necessary, enabling the user to enter numbers with a mouse rather than via keyboard entry.	No	"Public domain" (lack of originality)	The court noted that many other programs utilize a computer animated key pad, and such feature was therefore "not entitled to copyright protection because it was drawn from the public domain." The court also noted differences in the visual depictions when the keys were pressed.[511]
6. The program's short-hand method of entering distances using the mouse. Distances were entered in feet, inches, and sixteenths of inches. When either or both of the latter two was zero, the user could simply click the right button on the mouse, thereby entering zeroes without requiring the use of a keypad.	Yes		"The Court finds that the Aces Layout Programs' feature that permits the user to click the mouse in lieu of entering zero on the key pad, is original enough to warrant copyright protection."[512]

[509] It appears that the court's references to the "doctrine of public domain" are alternative nomenclature for a finding of lack of originality.

[510] Id.

[511] Id.

[512] Id.

7. Highlighting items selected by the viewer for greater identification and ease of use.	No	"Public domain" (lack of originality)	"The use of highlighting, like the key pad, is a common feature of computer programs. Therefore, the Court finds that it is a part of the public domain and not entitled to copyright protection."[513]
8. Expression of work lines in "UDLR" format – up-down left-right.	No	Merger and Scenes a faire	The terms "up," "down," "left" and "right" are "indispensable to a program aimed at designing structures, hence are unprotectable as *scenes a faire*. ... Moreover, the Court also finds that there is only one way to express the idea of direction in the context of a design program, therefore the idea and expression are considered to have merged."[514]
9. Description of planes using trapezoids, rather than with arrows or other methods.	No	Merger	"The Plaintiff's contention that instead of depicting planes visually as trapezoid shapes the Defendant could have used arrows, defeats the purpose of the intersecting plane system, which is to allow the designer to draft a truss plan three-dimensionally. Based upon the limited range of expression available to depict planes, the Court finds that the idea and expression have merged."[515]

[513] Id.

[514] Id. at 1581-82.

[515] Id. at 1582.

10. Expression of work planes without reference to the real walls, the roof, or the ceilings of the building. This feature made it easier to draw layouts of complicated structures.	Yes		"[A]lthough the doctrine of merger denies copyrightability to the depiction of planes as trapezoid shapes, the same cannot be said about the depiction of planes without reference to the connecting walls, ceilings or roof. The Court finds this is an original aspect of the Aces Layout Programs and is therefore entitled to copyright protection."[516]
11. The use of the word "cut" to refer to the intersection and boundaries of the planes.	No	Scenes a faire	The "Court finds that use of the term 'cut' in the context of designing roof trusses through the use of intersecting planes, is a term of art that is indispensable to the task. Accordingly, the Court finds that it is *scenes a faire*, and not entitled to copyright protection."[517]
12. Use of a reference line to position trusses.	Yes		"The Plaintiff argued at trial that the use of a 'reference line' in the center of the screen to position trusses, is a unique expressive aspect of the Aces Layout Programs and is not a step in the process a draftsman would undertake to design a roof truss layout by hand. The Court agrees."[518]
13. Use of a "rubber band line" to select portions of the drawings for certain editing functions. 14. Editing capabilities.	No	"Public domain" (lack of originality)	"Like the key pad discussed *supra*, the 'rubber band line' and rubber band box editing functions are common features found in many graphic design programs. Accordingly, the Court finds that its protectability is barred by the doctrine of public domain."[519]

[516] Id.

[517] Id.

[518] Id.

[519] Id.

15. Expressing 3-D views of the drawing by entry of two angles, rather than by entering distance, height, and view of the structure.	Yes		The "Court finds that the unique method the Aces Layout Programs employ to express its designs three-dimensionally, requiring the user to enter only two angles of the structure, is entitled to copyright protection."[520]
16. A four-box screen organization.	No	"Public domain" (lack of originality)	The Court found the four-box screen organization, including command menus along the top, down the sides and screen sizing functions along the bottom, to be common among a wide variety of programs. Thus the "use of a four-box, screen display is part of the public domain."[521]
17. Use of virtual memory to increase the capacity of the application software.	No	Not part of the program at all (idea or function)	"There was no evidence presented at trial indicating that the use of 'virtual memory,' or a portion of the program designed to increase the computer's memory to perform more complex designs, manifested itself visually in the user-interface of the program. Thus, the Court finds that it is not a nonliteral element of the program at issue in this litigation."[522]

[520] Id. at 1583.

[521] Id.

[522] Id.

18. Entry of walls in a "free" format, rather than just clockwise or counter-clockwise sequence.	Yes		"In considering the copyrightability of an element of a layout program, the proper inquiry is not whether other existing programs utilized that feature but whether that function is part of the logical series of steps a draftsman would undertake to create a plan. If the function falls within such a framework, then it is unprotectable as a process. ... The Court finds that there was no evidence presented at trial indicating that the placement of walls in a 'free' format is a part of the logical drafting process. Accordingly, the Court finds that this element is protectable."[523]

In sum, the court found that of the 18 elements identified by the plaintiff as allegedly copied, only five were entitled to copyright protection: (1) shorthand method of entering distances with the mouse, (2) expression of work planes without reference to real walls, (3) use of reference line, (4) entry of walls in free form and (5) expression of 3-D views of a drawing by entry of two angles."[524] It seems remarkable that virtually all of these 5 elements were really functional "features" of the program, most of which apparently had little visual manifestation in the user interface. Thus, the court seemed to conclude that the features themselves could be protected by copyright.

Having filtered the plaintiff's alleged list of similarities "to their core of protectable expression,"[525] the court then turned to the comparison step. With respect to this step, the court held that it must "look at the relative importance of the copied elements to the overall program to determine whether or not the Aces Layout Programs are substantially similar to the Arce Program."[526] As in the case of the Ninth Circuit's decision in Apple v. Microsoft, the court seems to have performed a comparison at both an individual element level and a work-as-a-whole level:

---

[523] Id. It is very curious why the court stated that the copyrightability of an element of a layout program does not turn on "whether other existing programs utilized that feature," because the court had, in considering the copyrightability of several other of the 18 allegedly similar elements (dynamic sizing, pop-up 15-key number pad, highlighting items selected by the viewer, rubber band line and rubber band box editing, and four-box screen organization), pointed to other graphics programs that had the element in question as a basis for ruling that such element was not protectable because of the "doctrine of public domain."

[524] Id. at 1584.

[525] Id.

[526] Id.

(a) Individual Element Comparison

With respect to comparison at an individual element level, the court ruled that, although there were similarities in four of the five protectable elements, the similarities were *de minimis*:

> The Court finds that of the five protectable elements identified in the Aces Layout Programs, four are substantially similar to *elements* in the Arce programs. A finding of a substantial similarity does not end the Court's inquiry, however. To find infringement, the Court must also determine that the Arce Program has appropriated substantial elements of the Aces Layout Programs. The Court has reviewed the programs and concludes that these five elements are not significant in the context of the Aces Layout Programs as a whole. [The five protectable elements] are not central to the operation of the Aces Layout Programs. They are instead *de minimis* and thus do not warrant a finding of substantial similarity.[527]

(b) Comparison of the Works as a Whole

With respect to a comparison of the works as a whole, the court, citing the Ninth Circuit's decision in Harper House, Inc. v. Thomas Nelson, Inc.,[528] held that, where a work consists "largely of uncopyrightable elements, as in the instant case, [they] are entitled to limited copyright protection," and infringement should not be found in the absence of "bodily appropriation."[529] Borrowing language very similar to that used in Apple v. Microsoft, the court held that "[t]he bodily appropriation standard requires a finding of virtual identicality."[530] The court ruled that the two programs were not virtually identical when viewed as a whole:

> The Court finds that even when viewing the Aces Layout Programs as compilations of uncopyrightable material, the programs are not substantially similar to the Arce Program. The Arce Program depicts its commands as icons in the Windows environment, rather than as words as in the Aces Layout Programs, thus there is no bodily appropriation of the entire visual display. ... As discussed above, the visual display of the Arce Program differs sufficiently, if not substantially, from the Aces Layout Programs to preclude a finding of virtual identicality. ... The dissimilarities which exist between the nonliteral elements of DOS and Windows programs generally, exist in the specific context of the Aces Layout Programs versus the Arce Program.[531]

---

[527] Id. (emphasis added).

[528] 889 F.2d 197 (9th Cir. 1989).

[529] Mitek, 864 F. Supp. at 1584.

[530] Id.

[531] Id.

In sum, the court concluded that the plaintiff's programs consisted mostly of elements that were unprotectable under the doctrines of merger, scenes a faire, and "public domain." Of the protectable elements that were substantially similar, the court found that their lack of importance in the context of the programs as a whole rendered any copying by the defendant to be *de minimis*. Accordingly, the court entered judgment for the defendant.

ANALYSIS OF THE ELEVENTH CIRCUIT'S DECISION

On appeal, the Eleventh Circuit, noting that the scope of copyright protection afforded to nonliteral elements of a computer program was an issue of first impression in the Circuit, rejected a number of challenges to the district court's decision, and affirmed. First, adopting the Tenth Circuit's approach in Gates Rubber, the Eleventh Circuit ruled that a claim of copyright infringement must be established by showing two things: (i) that the defendant, as a factual matter, copied portions of the plaintiff's program, and (ii) as a mixed issue of fact and law, that those elements of the program that have been copied are protected expression and of such importance to the copied work that the appropriation is actionable.[532]

On appeal, the plaintiff alleged that the district court failed to understand and apply the abstraction/filtration/comparison test of Altai correctly. Specifically, the plaintiff asserted that the district court erred by not performing a separate abstraction analysis because the plaintiff had provided a list of 18 non-literal elements that it claimed had been copied. Instead, the district court proceeded directly to the filtration step to determine whether such alleged similarities constituted copyrightable expression.

The Eleventh Circuit ruled, however, that it was not error for the district court not to further abstract the similarities at issue: "[I]f the copyright holder presents the court with a list of features that it believes to be protectable (i.e., original and outside of 17 U.S.C. § 102(b)), the court need not further abstract such features. Perhaps the best approach for a district court in any computer program infringement case, whether involving literal or nonliteral elements, is for it to require the copyright owner to inform the court as to what aspects or elements of its computer program it considers to be protectable.... After submitting a specification of the elements that it deemed to be protectable, MiTek cannot now argue that the district court failed to further abstract the elements of its own designation of protectable features."[533] This ruling by the court may influence other courts to require, as did Judge Walker in the Apple v. Microsoft cases, that the plaintiff provide the court with a detailed list of alleged similarities in copyright infringement cases. Having such a list in hand may also tend to influence courts to skip an independent application by the court of the abstraction step of the Altai analysis.

---

[532] Mitek Holdings, Inc. v. Arce Eng'g Co., 89 F.3d 1548, 1554 (11th Cir. 1996). Citing Gates Rubber, the Eleventh Circuit noted that proof of factual copying may be shown either by direct evidence, or inferred from indirect evidence by demonstrating that the defendant had access to the copyrighted work and that there are "probative similarities" between the allegedly infringing work and the copyrighted work. Id.

[533] Id. at 1555.

The plaintiff also contended on appeal that the district court, in failing to abstract the ACES program's menu and submenu command tree structure,[534] erred in concluding that it is an unprotectable "process" under §102(b). The Eleventh Circuit rejected this argument for two reasons. *First*, the court concluded – without a detailed analysis – that the ACES user interface was properly characterized as a "process" that could be protected only through the patent law: "If ... the patentable process and its expression are indistinguishable or inextricably intertwined, then 'the process merges with the expression and precludes copyright protection'.... Such is the case with the menu and the submenu command tree structure of the ACES program."[535] The Eleventh Circuit refused, however, to adopt the First Circuit's broad ruling in Lotus v. Borland, which the Eleventh Circuit read to hold that a menu command hierarchy is per se uncopyrightable: "Unlike the Lotus court, we need not decide today whether a main menu and submenu command tree structure is uncopyrightable as a matter of law. We agree with the conclusion reached by the district court that the ACES menu and submenu command tree structure is uncopyrightable under 17 U.S.C. § 102(b)."[536]

Second, the Eleventh Circuit held that the ACES program's particular menu structure was not protectable under the doctrines of originality and merger in view of the district court's finding that the program mimicked the steps a draftsman would follow in designing a roof truss plan by hand.[537] "[A]s a general matter, the idea of closely correlating the ACES program to the longhand steps taken by a draftsman was the constraining force in the design of the menu and submenu command tree structure. The logic inherent in this step-by-step process renders the resulting program unoriginal in that such logic may only be expressed in a limited number of ways."[538]

The plaintiff also contended on appeal that the district court erred in rejecting the plaintiff's argument that the user interface of its ACES program could be a protectable

---

[534] The Eleventh Circuit noted that a "'command tree' or 'command tree structure' informs the user, in a hierarchical fashion, of the options available, and also interacts with the user in requesting information from the user in order to utilize the program." Id. at 1556 n.18.

[535] Id. at 1556 n.19 (citing Atari Games Corp. v. Nintendo of America, Inc., 975 F.2d 832, 839-40 (Fed. Cir. 1992)).

[536] Id. at 1557.

[537] Id. at 1557 n.20. For similar reasons, the Eleventh Circuit rejected the plaintiff's challenge to the district court's characterization of the ACES program's use of trapezoids in truss design as a means of visually depicting planes. The plaintiff contended that the trapezoids were used not to depict planes, but rather to indicate to the user that a pitched or sloping plane for a particular wall had been defined. "We are not certain that the district court misconstrued the purpose behind the use of trapezoids, but even if it did, this use of trapezoids lacks sufficient originality to be entitled to copyright protection." Id. at 1557. Further, the Eleventh Circuit concluded that even if the plaintiff's use of trapezoids was protectable, the defendant's use of trapezoids in its programs "would constitute nothing more than nonactionable de minimis copying." Id. at 1557 n.22.

[538] Id. at 1558.

compilation when taken together as a whole. The Eleventh Circuit acknowledged that a user interface may be entitled to copyright protection as a compilation, and noted that it had never been established in its Circuit "what standard should be used in analyzing claims of compilation infringement of nonliteral elements of a computer program. Today, we join the Ninth Circuit in adopting the 'bodily appropriation of expression' or 'virtual identicality' standard."[539] Assuming without deciding that the nonliteral elements of the ACES user interface were a protectable compilation, the Eleventh Circuit agreed with the district court that there was no virtual identicality between the overall user interfaces of the plaintiff's and the defendant's programs.

Apart from the compilation claim, however, the Eleventh Circuit ruled that, if the district court applied the "substantial identicality standard in performing the comparison portion of the abstraction-filtration-comparison test, then it erred in doing so. Our circuit, in applying the Altai test, employs the substantial similarity standard in comparing what remains after the abstraction and filtration steps with respect to noncompilation copyrighted works."[540] The Eleventh Circuit concluded, however, that the district court apparently used imprecise language in an earlier portion of its opinion, but nevertheless had later correctly compared for substantial similarity in reaching its ultimate conclusion. Accordingly, the Eleventh Circuit rejected the plaintiff's contention that the district court erred in applying the wrong standard of similarity.

Finally, in a very brief portion of its opinion, the Eleventh Circuit rejected the plaintiff's contention that the district court had erred in concluding that any similarities between the programs at issue were de minimis. "We agree with the district court that the elements that were considered original and appropriated were not of such significance to the overall program to warrant an ultimate finding of substantial similarity and hence infringement. The burden is on the copyright owner to demonstrate the significance of the copied features, and, in this case, MiTek has failed to meet that burden."[541]

O. THE BAYSTATE CASE

One of the most important "feel" cases decided in 1996 is that of Baystate Technologies, Inc. v. Bentley Sys.[542] This case is another one dealing with the very important issue of the copyrightability of a computer program "technical" interface – i.e., the interface through which the software or firmware of two programs or devices exchange information with each other, such as through operating system calls, parameter structures, and input/output formats. Technical interfaces typically are comprised of a set of command terms and accompanying rules that must

---

[539] Id. at 1558 (citing Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1446 (9th Cir. 1994) and Harper House, Inc. v. Thomas Nelson, Inc., 889 F.2d 197, 205 ((9th Cir. 1989)). The Eleventh Circuit noted that both terms convey "a level of similarity greater than the 'substantial similarity' standard of the Altai abstraction-filtration-comparison test." Id. at 1558 n.24.

[540] Id. at 1559.

[541] Id. at 1560.

[542] 946 F. Supp. 1079 (D. Mass. 1996).

be observed in order to communicate through the interface. "Copying" of a technical interface usually occurs, as in Baystate, as a result of the need for compatibility with, or interoperability with, another program or device utilizing the technical interface.

Baystate is an especially important case because the protectability of technical interface elements required to create compatible works was clearly posed and decided, and the opinion is more thoroughly reasoned than many of the other decisions that have dealt with technical interface compatibility issues.[543]

1. Factual Background

The plaintiff, Baystate, was the owner of the copyright in a CAD program known as CADKEY. Baystate acquired ownership of the rights to CADKEY from its original developer, Cadkey, Inc. The defendant, Bentley, marketed a competing CAD program known as Microstation. At issue in the case was a data translator program marketed by Bentley that would translate data files between CADKEY and Microstation. Bentley hired an outside third party, Infotech, to develop the translator.

Infotech was selected to develop the translator because Infotech had previous experience at developing a program that was capable of reading CADKEY files into a CAD program of Infotech known as MODES (an acronym for Management of Drawings and Engineering Systems). To develop this "read" capability for MODES, Infotech used a copy of the documentation for the CADKEY Part File Tool Kit (the "Tool Kit") that it had received from the former President of Cadkey, Inc. The Tool Kit documentation described the organization of the CADKEY file data structures and file formats, and the access functions that were included in the library of executable files for CADKEY. Baystate held a registered copyright in the Tool Kit documentation.

At the time Bentley retained Infotech to develop the translator program, Infotech was also doing development work on CADKEY under contract with Cadkey, Inc. As a result of its

---

[543] Although the case was appealed to the First Circuit, the parties reached a settlement during the pendency of the appeal. Unfortunately, then, no opinion from the First Circuit was ever rendered in the case. Two other cases discussed in this paper indicated that technical interfaces may not be protectable: Bateman v. Mnemonics, Inc., 79 F.3d 1532 (11th Cir. 1996); Mitel Inc. v. Iqtel, Inc., 896 F. Supp. 1050 (D. Col. 1995). In addition, in the earlier case of Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992), the Ninth Circuit held that Accolade's intermediate copying through disassembly of Sega's object code in order to glean the functional requirements for compatibility with the Sega Genesis game console constituted a fair use. The court ruled that the functional requirements for compatibility with the Genesis console were aspects of Sega's programs that were not protectable by copyright. Id. at 1522.

By contrast, three other cases discussed in this paper indicated that technical interfaces may be protectable by copyright: Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994); Control Data Sys. v. Infoware, Inc., 903 F. Supp. 1316 (D. Minn. 1995); and CMAX/Cleveland, Inc. v. UCR, Inc., 804 F. Supp. 337 (M.D. Ga. 1992).

contract with Cadkey, Inc., Infotech had in its possession proprietary information and confidential source code relating to CADKEY. Infotech therefore created a "wall" between those programmers who were working on CADKEY and those programmers who were working on development of the translator for Bentley. The court ruled, however, that this "wall" crumbled in at least two respects. First, an engineer of Infotech supervised both the translator project and the work that Infotech was doing for Cadkey, Inc., and one programmer worked both on the translator project and did some minimal work on the project for Cadkey, Inc. Second, and more significantly, the court noted that Infotech admitted that the "read" portion of its MODES program (which had been developed with reference to the Tool Kit documentation) was used in developing part of the translator.[544]

In August of 1996, Infotech sent by e-mail to Bentley a portion of the source code for the translator it was developing for Bentley. Infotech mistakenly copied the e-mail to Baystate (Baystate had succeeded to the contract between Infotech and Cadkey, Inc. when Baystate purchased CADKEY from Cadkey, Inc.). Baystate analyzed the unsolicited source code for the translator program and came to the conclusion that source code from the CADKEY "header files" had been copied by Infotech into the translator program. The "header files" were a set of definitional files that defined the data structures used by the CADKEY program.

After Bentley began advertising that it would shortly release a CADKEY-to-Microstation translator, Baystate filed a copyright suit against Bentley. Baystate asserted copyright rights in three works: the CADKEY source code, the Tool Kit source code, and the Tool Kit documentation. Baystate argued that Bentley's translator was infringing because it copied the data structures defined in the CADKEY header files and used in the CADKEY program itself. Other than the similarity in the data structures, Baystate did not assert that there was any similarity between the source code or object code of the CADKEY program and the translator program.

2. Legal Analysis

The district court began its legal analysis by ruling that, although the CADKEY computer program itself was copyright protected, the data structures themselves were not copyrightable as computer programs per se: "because the data structures at issue in this case do not bring about any result on their own, they are copyright protected, if at all, only as a part of the whole computer program."[545]

The court noted that, under the First Circuit's test for copyright infringement, Baystate was required to establish factual copying of its copyrighted work through proof of access plus probative similarity between the protected work and the allegedly infringing work. The district court ruled that Baystate had established such factual copying: Infotech had access to the CADKEY source code, the Tool Kit source code, and the Took Kit documentation, and the "wall" Infotech had attempted to maintain between the translator project and the project for Cadkey, Inc. had crumbled for the reasons noted above. The court also concluded that there

---

[544] Baystate, 946 F. Supp. at 1085, 1087.

[545] Id. at 1086.

existed a probative similarity "between the names and organization of the data structures of the Part File Tool Kit source code and documentation, on the one hand, and the Translator as transmitted in the e-mail of August 21, 1996, on the other hand."[546]

The court ruled, however, that the allegedly copied elements – the data structure names and the organization of the files within the data structures – were not protected by copyright, nor were they constituent, original elements of the copyrighted CADKEY program. In determining whether the data structure names and organization of files were copyrightable, the court adopted the abstraction-filtration-comparison test applied in Gates Rubber.

With respect to the allegedly copied data structure names – "or, more specifically, the words and abbreviations used to describe the files contained within the data structures"[547] – the court ruled, citing the First Circuit's Lotus v. Borland decision, that they were unprotectable under the merger doctrine because the names were typically related to the function of the file.[548] The court's analysis was confined to the merger of individual names with function, and the court did not consider whether the names taken as a collective whole might have avoided the merger doctrine. The defendant would no doubt have argued, however, that there was no independent "selection" or "arrangement" of the names that could constitute protectable expression in the compilation of names. With respect to selection of names, one could argue that the collection of names as a whole was no more than the sum of its individual parts. With respect to arrangement of the names in the data structure, one could argue that the arrangement was itself functional (i.e., merged with function) because the arrangement of elements within a data structure is the very essence of the structured data, and is the aspect of the data that enables it to be efficiently referenced and processed by the program utilizing it.[549]

With respect to the organization of the data structures, the court ruled that they were unprotectable under the scenes a faire doctrine because of considerations of compatibility and efficiency:

---

[546] Id. at 1087.

[547] Id. at 1088.

[548] Id. Although this holding appears to be case specific, it would probably apply to many technical interfaces, for command names and parameter names are often expressed in descriptive, mnemonic terms.

[549] Cf. Zimmerman, "*Baystate* Holding: Technical Interfaces Not Copyrightable – On to the First Circuit!", 14 The Computer Lawyer 9, 16 (April 1997): "[A]t least with regard to the order in which the names are arrayed within any particular data structure ... the arrangement of names can reasonably be said to be merged into the arrangement of substantive elements comprising the structure itself which the arrangement reflects. In other words, the programmer is likely to have exercised judgment in setting forth the arrangement of variables or other elements that make up a data structure or its substructures, or to have employed a syntax that may or may not be original. Having made those judgments and employed that syntax, the arrangement of the names given to those elements simply follows from (and should be deemed merged with) that underlying determination of the structure of data elements and sub-structures and the syntax employed."

> In this case, the court concludes that the selection and organization of the elements in the data files is dictated mainly by external factors. The product being developed is a data translator that is designed to "read" the data files of CADKEY. The process of "reading" the CADKEY data files requires that the elements contained within the data structures of the Translator be organized in the same manner as the elements in the data structures of CADKEY. Without such compatibility, the Translator would not function because it would "misread" the CADKEY data files.
>
> Additionally, the organization of names is, at least partly, a function of efficiency. Walter Anderson, defendant's expert witness, testified that the names and the arrangement of those names serve a functional and necessary purpose in the code of a data translator. They permit a computer programmer who produces a data translator to refer back to the documentation of the "target product" (the program the data translator seeks to read) as he creates the code for the translator.
>
> Significant differences between the names and organization of the names used in the "target product" and the translator would be inefficient for the programmer. For that reason, the data structure names in Infotech's MODES product had to be similar to the data structures in the Part File Tool Kit documentation because the computer programmer needed to refer to the documentation in the process of creating and manipulating the CADKEY "read" capability of MODES.[550]

In addition, the court noted that the Tool Kit documentation itself had stressed the necessity of following the CADKEY structures closely: "It is up to the programmer to comply with the CADKEY entity structures and to codify and enforce an internal structure for any user defined entities or else chaos will ensue!"[551]

The court's application of the scenes a faire doctrine is a bit curious. Although the court states that the selection and organization of elements in the data files was "dictated mainly by external factors,"[552] the court is making this statement from the viewpoint of the defendant, or the second comer, who desires to be compatible with the data structures selected by the plaintiff. There was no evidence in the case that the plaintiff's original choices of names or organization of files within the data structures were dictated by any externality, which is traditionally the issue to

---

[550] Baystate, 946 F. Supp. at 1088-89.

[551] Id. at 1089. The court further invoked "industry-wide standards" in noting that two witnesses who had developed translators to read the data files of other CAD programs (including CADKEY) testified that they were required to use the file names and the organization of the data structures of the "target program" in their translator. Id. This is likewise an unusual application of the "industry standards" limiting doctrine, which traditionally would look to whether the plaintiff's actual data structures had become an industry standard, not whether copying of data structures in general was an industry practice.

[552] Id. at 1088.

be addressed in determining whether to apply the scenes a faire doctrine. However, those names and organization having been selected by the plaintiff, it was necessary, just as in the Lotus v. Borland case, for the defendant to "copy" them in order to be able to read files created by CADKEY using those names and file organization. The court used this fact to invoke the scenes a faire doctrine, which is an unusual application of that doctrine. A more conventional basis for the court's holding would, for example, have been to invoke the fair use doctrine to justify the copying, for the fair use doctrine traditionally looks to the actions of the defendant in justifying its copying, rather than to whether external constraints dictated the plaintiff's original choices in its copyrighted work.

Finally, turning to the issue of infringement, the court ruled that, even if the data structures at issue were protected under the copyright laws, those parts of the computer program constituted neither a substantial portion nor a significant aspect of the whole copyrighted work for the following reasons:

> The expert testimony showed that, although data structures are generally a necessary component of a computer program for organizational and efficiency purposes, the original naming of data structures takes very little of the total time or creative genius necessary to develop a program. Furthermore, data structures are not, by themselves, executable, i.e. a computer cannot read data structures and perform any function. Although the importance of a program component is not strictly a function of quantity, the evidence in this case demonstrates that the subject data structures represent only a small portion of the total CADKEY program.[553]

The court also addressed a claim by Baystate that the copyright in its Tool Kit documentation was infringed, because the documentation contained the data structures that Baystate alleged were copied. Although the court noted that part of the translator source code was similar to parts of the Tool Kit documentation, "[t]hat documentation is not, however, at all similar to the Translator in an overall comparison of content, purpose or use." The court therefore concluded that there was no substantial similarity between the documentation itself and the translator program. Citing Baker v. Selden,[554] the court also ruled that "Baystate's copyright of the Part File Tool Kit documentation does not grant it an exclusive right to use the information conveyed in that documentation and thus, Infotech's use thereof does not infringe that copyright."[555]

Although earlier precedents have reached seemingly contradictory conclusions, the Baystate case affords an important precedent for those who would copy header files, data structures, file structures, or similar structural elements, where such copying is necessary for reading the data output by a program, for supplying data in the input format required by another program, or for otherwise achieving interoperability with another program. The Baystate case

---

[553] Id. at 1090.

[554] 101 U.S. 99 (1879).

[555] 946 F. Supp. at 1090.

also suggests, following the reasoning of Baker v. Selden, that those desiring to achieve such interoperability are entitled to glean the structural elements necessary for compatibility from a copyrighted manual or other documentation describing such elements and then to create a functional implementation of the described elements.

P. THE HARBOR SOFTWARE CASE

Another case decided in 1996, Harbor Software, Inc. v. Applied Sys.,[556] afforded an opportunity for the court to address how to apply the abstraction/filtration/comparison test of Altai in the context of a jury trial. The plaintiff was the owner of a computer program called the "Sales Center Manager" (SCM), which was designed to provide automated marketing services to insurance agencies, such as client development and management, direct mailing campaigns, client followup, and calculation of various statistics tracking client activity and business development. The plaintiff alleged that the defendant's competing product, known as "The Agency Manager," infringed the plaintiff's copyright in SCM by copying several non-literal aspects of SCM.

The court adopted the abstraction/filtration/comparison analysis of the Altai case. Because the case was to be tried to a jury, the court ruled that the case should be divided procedurally into two phases. Analogizing to the Federal Circuit's decision in Markman v. Westview Instruments, Inc.,[557] which held that patent claims should be construed by the court as a matter of law to define the scope of the patent, the court in Harbor Software concluded that the abstraction and filtration analysis should be applied by the court in the first phase to define the scope of the plaintiff's copyright. Then, the second phase would be comprised of consideration of motions for summary judgment and "a jury trial on the comparison of nonliteral elements found to be protectable expression."[558]

The court then made various rulings under the abstraction and filtration analysis to define what similarities alleged by the plaintiff constituted protectable expression. The court had appointed an expert to assist in this process. The expert worked with the parties during the abstraction process to develop a series of exhibits that graphically represented the nonliteral elements of SCM for which the plaintiff sought copyright protection. The expert also made his own recommendations to the court as to which elements of SCM should be protectable based, for example, on his views of whether the alleged similarities constituted common programming techniques or abstract functions or ideas, and whether there were other ways to write code or to organize modules or data flow in order to perform the functions at issue.

The court used these exhibits to make its rulings under the filtration analysis as to which alleged similarities constituted protectable expression. In creating the exhibits, the plaintiff began at the architectural level of abstraction and roughly followed one or another of the levels of abstraction set forth in the Gates Rubber case. The defendant objected to the fact that the

---

[556] 936 F. Supp. 167 (S.D.N.Y. 1996).

[557] 52 F.3d 967 (Fed. Cir. 1995), aff'd, 116 S. Ct. 1384 (1996).

[558] Id. at 1046.

abstractions in the exhibits did not correspond exactly to various defined modules of code in the program, but rather were based on the functional aspects of the code. The defendant argued that Altai focused on modular structure and therefore required that the abstractions conform to the modular structure of the program. The court rejected this argument, noting that Altai discussed not only modular structure, but also the inter-relationships between modules. "These relationships can also be classified as control flow and data flow. Thus plaintiff has focussed [sic] on the data flow and control flow of its program rather than its strict modular structure."[559]

Turning to its filtration analysis of the exhibits, the court concluded that the following similarities did not constitute protectable expression:

- Similarities in structure and data flow at the highest level of program architecture, because of merger. "Insufficient alternatives exist for programmers to create expressions at lower levels that would not appear to be identical at this level of abstraction."[560]

- The control flow for automated prospecting, which the court found had merged with "the process of automated prospecting."[561]

- The general control and data flow for initiating a prospect or client on a marketing plan, which the court found had merged with the process described.[562]

- The file locking steps of SCM, which the court found to be contained in computer programming books and therefore constituted standard techniques in the public domain.[563]

- The data flow and control flow for searching an individual prospect/client and modifying prospect/client information, which the court found to be unprotectable methods of operation.[564]

- The control flow permitting a user to (i) interact with the program and search and modify prospective and current client information and (ii) entering prospect data and performing searches on the prospect/client file, both of which the court found to be merged with the user process described.[565]

---

[559] Id. at 1047.

[560] Id. at 1048.

[561] Id. at 1049

[562] Id.

[563] Id. at 1051.

[564] Id.

[565] Id.

- The algorithm for calculating cyclical statistics, which the court found to be an unprotectable method of operation, although the court did rule that certain "expressive choices made by the programmer as to the particular steps taken in calculating cyclical statistics," despite being "close to the merger line," were protectable.[566]

By contrast, the court ruled that the following similarities, among others, were protectable expression:

- The detailed data flow relationships involved in how information processed in the automated prospecting module progressed through SCM.[567]

- The detailed structure of "the process by which a user initiates a marketing plan."[568]

- The selection and organization of the database fields in the SCM data structures.[569]

- The "relationships between the modules and database files" in the self-administered marketing module, which the court found were not dictated by efficiency or external limitations.[570]

- The "organization of the three searches" implemented in the self-administering marketing module, which the court found not dictated by efficiency or external factors.[571]

- The selection and arrangement of the information included in SCM's various screen reports and displays, which the court found to be a protectable compilation.[572]

Following the court's rulings, the defendant moved for summary judgment of noninfringement. In a subsequent opinion ruling on the defendant's motion,[573] the court was called upon to decide the standard of similarity that would be applicable in the comparison analysis. The court noted that three levels of scrutiny could potentially be applied in the comparison step: (i) The "ordinary observer test," which the court noted is the standard test for substantial similarity, and which asks whether an average lay observer would recognize the alleged copy as having been appropriated from the copyrighted work; (ii) A "more discerning"

---

[566] Id. at 1052.

[567] Id. at 1049.

[568] Id.

[569] Id.

[570] Id. at 1050.

[571] Id. at 1051.

[572] Id. at 1052.

[573] Harbor Software, Inc. v. Applied Sys., 936 F. Supp. 167 (S.D.N.Y. 1996).

version of the ordinary observer test, which is applicable where the copyrighted work contains both protectable and unprotectable elements, and with respect to which the court "must attempt to extract the unprotectable elements from our consideration and ask whether the protectible elements, standing alone, are substantially similar"; and (iii) The "trivial difference" standard, which is applicable to works in which the author contributed only a minimum of creativity, such as a fact-based compilation, with respect to which a defendant need only demonstrate that its work "differs in more than a trivial degree" from the plaintiff's work in order to defeat an infringement claim.[574]

The defendant contended most of the exhibits should be compared under the "trivial difference" standard, whereas the plaintiff argued that the traditional substantial similarity test was appropriate for all comparisons. The court ruled that the "trivial difference" test should be applied to the screen displays and reports, since they were compilations of factual information, such as names, addresses, insurance industry codes, and insurance marketing statistics. All other similarities were, however, to be judged under the substantial similarity test.[575]

Turning to the defendant's motion for summary judgment, the court denied the motion in part and granted it in part. The court found that with respect to the majority of alleged similarities the court had ruled protectable, there were genuine issues of material fact as to whether there was substantial similarity between the two programs, and summary judgment as to those similarities must therefore be denied.[576] The court concluded, however, that with respect to certain alleged similarities in various of the reports, the defendant's reports differed by more than a trivial degree – and were also not substantially similar under the traditional ordinary observer test – and therefore were not infringing as a matter of law. The court based its conclusion on the fact that the plaintiff's reports were organized by horizontal row according to date, whereas the defendant's reports were organized by horizontal row according to marketing event, and the selection of data categories and the arrangement of the reports was different between the plaintiff's and defendant's programs.[577]

IV. CONCLUSIONS FROM THE "LOOK" AND "FEEL" CASES

The courts have struggled mightily during the last few years to draw boundaries around what elements of a computer program should be considered protectable expression, and what elements should be unprotected under various traditional "limiting" doctrines – such as the idea/expression distinction, the merger doctrine, scenes a faire, and the limitations imposed by § 102(b) of the copyright statute.

In their efforts to establish the boundaries of protection, the courts have devised a number of tests, either for copyrightability or infringement, or both. All of these tests have their

---

[574] Id. at 170.

[575] Id. at 171.

[576] Id.

[577] Id. at 171-72.

fundamental roots in the traditional two-step test for copyright infringement (variously called the "extrinsic/intrinsic test," the "objective/subjective test," and the "Arnstein test"[578]) – the first step of which consists of analytic dissection of ideas and similarities in the works at issue (to determine "copying" and whether the copied elements are protectable), and the second step of which consists of some form of overall judgment of substantial similarity (to determine "illicit copying"), traditionally under a "lay observer" test.

As applied to computer programs, however, this traditional two-step test has been transmogrified and glossed in many ways by the courts in the various Circuits. The result has been that the scope of protection differs Circuit by Circuit, and the cases are sometimes difficult to reconcile. Nevertheless, a few observations can be drawn from the landscape that is forming out of the several cases analyzed in this article. Although these observations cannot be regarded as firm conclusions, and should be taken as merely an attempt to put some conceptual gloss over what is admittedly an evolving picture, the exercise seems worthwhile.

A. THE SCOPE OF PROTECTION VARIES BY CIRCUIT

Perhaps the most obvious conclusion to be drawn is that under the current decisions, the scope of copyright protection for computer programs varies by Circuit. Protection has been narrowed in the First, Second, Ninth and Tenth Circuits. Under the abstraction/filtration /comparison test of the Second Circuit in Altai and of the Tenth Circuit in Gates Rubber, at least with respect to nonliteral elements of computer program code, many elements will probably be filtered out of the analysis as limited by considerations of efficiency or as dictated by the broad list of "external factors" recognized by those courts. Similarly, although the Eleventh Circuit has not explicitly adopted the abstraction/filtration/comparison test, in the Bateman case it implicitly accepted a district court's use of that test, and stated in dicta that copying for compatibility purposes might be justified under particular circumstances by virtue of various limiting doctrines of copyright law.

Similarly, under the Ninth Circuit's objective/subjective test of Brown Bag and Apple v. Microsoft, analytic dissection of similarities plays a significant role, and the Ninth Circuit has exhibited willingness in Brown Bag, Apple v. Microsoft, and the earlier Data East v. Epyx decision to classify many user interface elements as unprotectable, at least standing alone. Although the First Circuit's decision in the Borland case does not adopt the abstraction/filtration/comparison approach, it rejects copyright protection for computer program menu command structures, and its definition of a "method of operation" under § 102(b) of the copyright statute may render other functional aspects of a computer user interface not protectable. [579]

---

[578] Arnstein v. Porter, 154 F.2d 464 (2d Cir. 1946).

[579] But cf. Williams v. Arndt, 626 F. Supp. 571 (D. Mass. 1985), in the First Circuit's jurisdiction, which afforded broad protection to a method of floor trading. In that case, the plaintiff alleged that the defendant's programming of a floor trader's method described in the plaintiff's copyrighted manual infringed the copyright in the manual. The court was heavily influenced in its finding of infringement by an in-court demonstration given by the plaintiff's expert, who performed the calculations set out in the manual by hand, using data from The

By contrast, the Whelan decision still sets forth the governing rule in the Third Circuit, and that decision's definition of "idea" at the highest level of abstraction affords potentially very broad protection. In the Fifth Circuit, although the Court recently adopted the abstraction/filtration/comparison test in the Engineering Dynamics case, it applied that test leniently in the filtration step in construing what elements of computer program input formats might be protectable, notwithstanding its earlier decision in the Plains Cotton case, which recognized market "externalities" as limiting the scope of copyright protection.[580] In the D.C. Circuit, the Atari v. Oman decision suggests that even the most simple of computer program user interfaces can be protected by copyright, although the opinion intimates that such protection would be quite "thin" for such simple works.

In addition, district court decisions in the Sixth Circuit (Consul Tec) and the Eleventh Circuit (CMAX, Softklone and Mitek) have afforded broad protection to nonliteral elements of computer programs, although the Bateman decision from the Eleventh Circuit may signal an openness to limiting the scope of copyright protection in appropriate circumstances where copying has been done for compatibility. In addition, the Tenth Circuit seems to have applied its abstraction/filtration/comparison test to afford much broader protection in the Autoskill case than it did in Gates Rubber, leaving somewhat murky the scope of copyright protection that the Tenth Circuit will ultimately afford to computer programs.

B. "LOOK" HAS CONTRACTED, WHILE "FEEL" HAS EXPANDED

Looking at the various decisions as a group, without regard to the Circuits from which they originate, it appears that over the last few years, the scope of "look" protection has generally contracted, while the scope of "feel" protection has expanded in many areas (although during 1995 more decisions began to allow copying for various compatibility reasons). All of the prominent "look" cases in the Ninth Circuit (Data East v. Epyx, Brown Bag and Apple v. Microsoft) have rejected broad claims by the plaintiff. The "limiting" doctrines of copyright law have been applied with vigor in the dissection process to eliminate most or all of the alleged similarities from the calculus of infringement. More recently, in the Productivity Software case, the court rejected a claim for infringement of the overall interface of an add-in program to expand short form abbreviations to long forms, ruling that all similarities between the plaintiff's and defendant's programs related only to uncopyrightable features and scenes a faire.

If the abstraction/filtration/comparison test of the Second and Tenth Circuits is applied to "look" cases – despite the Second Circuit's initial statement that the test was not intended for those types of cases and the fact that the Tenth Circuit's adoption of that test in Gates Rubber was primarily in the context of a "feel" case – one may expect "look" protection to narrow in those Circuits as well. Given the historically persuasive force of the Second and Ninth Circuits

---

Wall Street Journal. The expert then ran the defendant's program using the same data, and the result was the same. See id. at 580-81.

[580] Plains Cotton Cooperative v. Goodpasture Computer Serv., 807 F.2d 1256 (5th Cir.), cert. denied, 484 U.S. 821 (1987). See supra note 20 and accompanying text.

in copyright matters, the trend of these two Circuits to narrow look and feel protection is significant for future directions (as has already been manifested in the Tenth Circuit).

By contrast, a significant number of decisions in the last four years – including the Fifth Circuit's decision in Engineering Dynamics, the Tenth Circuit's decision in Autoskill, and several district court decisions (Consul Tec, CMAX and Mitek) – have expanded the scope of protection for nonliteral structural and "feel" elements of programs. Computer input formats have been ruled potentially protectable in the Engineering Dynamics case, and the district court in the Interactive Network case held that the arbitrary ordering and structure of the data fields in the plaintiff's "data feed format" for interactive video games was protectable subject matter. Command syntax seems to have been protected in the Consul Tec case.

"Internal" structural program elements such as file structures, constants and alphanumeric codes have been protected in the Consul Tec and CMAX cases. At least two cases – Consul Tec and Autoskill – have protected the operational flow control and/or underlying methodologies of a program. The Mitek case ruled that five features – most of them functional – could be protected by copyright, although the court concluded that the defendant's copying of such features fell within the *de minimis* doctrine.

It should be noted, however, that many of the decisions expanding "feel" protection have been issued by district courts. If the Borland decision is any portender, the trend to expand "feel" protection may begin to reverse itself as more cases in the "feel" area reach the appellate courts. As is evident in several of the appellate decisions discussed in this article, the courts seem to be increasingly sensitive to the risks of overprotection of functional elements of computer programs and willing to apply the various limiting doctrines of copyright law to a broader number of aspects of computer programs. Thus, one may see the same trend in the next few years to reduce "feel" protection that has already been manifested in the "look" cases, particularly as more cases reach the appellate courts.

As evidenced by decisions issued in 1995, much of the action in future "feel" cases will probably concern "copying for compatibility." The courts have varied widely in what they have permitted to be copied in the name of compatibility. In the Mitel case, Mitel's command codes for programming telephone call controllers were held uncopyrightable and the defendant was permitted to copy them in a competing controller.

By contrast, the Eleventh Circuit in the Bateman case held that interface commands are not per se uncopyrightable, although it noted in dicta that under appropriate circumstances copying of the same might be permissible under any of several limiting doctrines. More recently, threshold parameter values for determining when a hard disk drive is about to fail were held protectable in the Compaq case; these values could not be copied onto compatible disk drives. In the Control Data case, with very little analysis, the court issued a preliminary injunction against the developer of a program that emulated a network operating system.

Thus, at the present time, the scope of what may be permissibly copied to achieve compatibility is considerably uncertain.

C. PURE "CLONERS" ALWAYS LOSE

Before the First Circuit's decision in the Borland case, one "rule" which had emerged from the cases was that pure "cloners" – those who set out to make identical copies or "clones" of a computer program or a substantial part thereof – always lose, regardless of what elements are cloned. See Softklone, Paperback, Consul Tec, and CMAX. A corollary "rule" that may be implicit in the recent "feel" cases at the district court level is that there is a good chance that programs designed to emulate the operational "behavior" of another program in close detail – regardless of whether the two programs "look" alike – will be ruled infringing. See Control Data and Interactive Network.

Since the Borland decision, this "rule" has eroded somewhat. For example, the Mitel case permitted a "cloner" to copy the command codes of the dominant industry supplier. And the Bateman case noted in dicta that the interface commands to an operating system might, under appropriate circumstances, be copied under various limiting doctrines of copyright law.

D. THE INFRINGEMENT TESTS ARE NOW MORE REFINED

Although the courts have come up with a number of tests for judging copyrightability, substantial similarity and/or infringement generally, virtually all courts outside the Third Circuit have uniformly rejected the test of Whelan for what constitutes expression in a computer program in favor of more sophisticated tests. The Whelan rule has been widely recognized as overly simplistic and potentially overbroad. There is growing agreement that a computer program may contain many "ideas" at many levels, and the courts are attempting to apply an abstractions test – most recently in the "filtration" step (known as the "intrinsic" test in the Ninth Circuit) – in order to decide what should be excluded from the calculus of infringement as unprotectable. One might expect as an initial matter that broader application of an abstractions test would narrow the scope of what is protectable down from that of the Whelan rule, although the "feel" cases have demonstrated that courts can afford very broad protection even under the abstractions test.

E. THE TREND IS TOWARD FILTRATION

A component element consistently appearing in the various forms of emerging infringement tests is that of "filtration" or "dissection" of a copyrighted work to determine which elements are unprotectable. The Shaw case out of the Ninth Circuit placed a dissection/filtration element into the "extrinsic" step of its infringement test, and the Brown Bag case exported that approach to computer programs. Filtration became the central analytical focus of the Apple v. Microsoft case, both in the district court and in the Ninth Circuit, and of the district court's decisions in the Capcom v. Data East case. The Altai case in the Second Circuit, the Gates Rubber case in the Tenth Circuit, and the Engineering Dynamics case in the Fifth Circuit make filtration the explicit second step of the infringement analysis. Although the Second Circuit stated in Altai that its test was intended for nonliteral structural cases, courts in the Tenth Circuit (Gates Rubber and Autoskill) have seemed willing to apply that test or extracts thereof to analyze "look," as well as "feel," elements in user interfaces.

To date, however, the question of what happens post "filtration" – that is, can filtered elements still form part of the analysis of overall substantial similarity – has continued to present a conundrum in the cases. The "feel" cases have been more definite in their pronouncements on the issue. At least three "feel" cases (Altai, Gates Rubber, and Autoskill) seem to establish that

the filtered elements must be completely disregarded, and substantial similarity must be judged only with respect to what remains after the filtration.

By contrast, the "look" cases leave largely unanswered the question of what one is to do with elements of a user interface that, standing alone, may be unprotectable, but may form part of some larger totality that is greater than the sum of its parts. The Apple v. Microsoft, Brown Bag and Capcom v. Data East decisions leave open the possibility that "unprotectable" elements may nevertheless form part of some "larger" arrangement, sequence, selection or layout of elements that may constitute protectable expression taken as a whole. Those decisions do not, however, specify how such unprotectable elements are to be handled, particularly when such elements are to be presented to a jury. Conversely, where all similarities are unprotectable, and there is no "larger" expression formed of unprotectable component parts, the Data East v. Epyx and Brown Bag cases hold that the defendant is entitled to a ruling of noninfringement as a matter of law.

Unfortunately, this unanswered question concerning what one does with "filtered" elements is one of the most important questions for determining the scope of protection that courts will ultimately afford to "look and feel" claims. Pending further judicial development and explication in the courts, those contemplating copying visual (and perhaps other) elements of an interface should not assume that infringement can be avoided merely because the copied elements may be subject to various "limiting" doctrines when considered apart from other elements of the total interface.

F. THERE IS GREATER RECOGNITION OF "EXTERNALITIES"

In applying the filtration process, the decisions now recognize more "externalities" that may limit the scope of protection and call for filtration of an element from the similarity analysis. During the last few years, the courts have increasingly recognized "compatibility" and "standardization" as possible externalities, although the scope of these doctrines is still ill defined, and there continue to be many decisions that reject arguments based on these notions. Thus, at this point, a potential defendant cannot predictably rely on some of the newer limiting doctrines that are beginning to emerge.

A cataloging of some of the externalities that recent decisions have mentioned is as follows:

- Hardware constraints (Altai, Gates Rubber, and CAMS)
- Functionality – purely functional items or an arrangement of them for functional purposes (Apple v. Microsoft and Capcom v. Data East)
- Standardization (Gates Rubber, Apple v. Microsoft, Brown Bag, Capcom v. Data East, Mitel, and Productivity Software)
- Expectations of users (Apple v. Microsoft, Brown Bag, Altai, Capcom v. Data East and Mitel)
- Compatibility requirements (Altai, Gates Rubber, Borland, Bateman, and Mitel)

- Efficiency (Altai, Mitel and Productivity Software)
- Computer manufacturers' design standards (Gates Rubber and Altai)
- Widely accepted programming practices (Altai and Gates Rubber)
- Elements that have entered the public domain through free accessibility (Altai and Mitek)
- Target industry practices and demands (Gates Rubber)

G. CERTAIN STANDARD USER INTERFACE ELEMENTS ARE UNPROTECTABLE

Despite a wide range of ultimate outcomes in the look and feel cases, there seems to be a growing consensus emerging from the analysis of the various interfaces in the cases that certain standard user interface elements are unprotectable, and may not form a basis for infringement:

- The fundamental functional features of a graphical user interface, apart from their specific implementation or appearance – overlapping windows, iconic representation of objects, object opening and closing, menus and iconic manipulation.
- Use of pull down windows in the menu system.
- Use of the space bar, backspace key, cursor keys or other standard keys to navigate among menu items or to activate a menu.
- Use of the return key to select an item.
- Use of the +, -, * and / keys to represent the mathematical operations of addition, subtraction, multiplication and division.
- Commands or menu items that are standard to all programs of a particular genre – such as "Cut," "Copy," and "Paste" in a word processing program.

- Outlining of lists with a single line border
- Use of one screen for data entry and a separate screen for editing data

H. THERE IS AN INCREASING USE OF EXPERT TESTIMONY TO JUDGE INFRINGEMENT

Courts now widely allow use of expert testimony in judging infringement, both at the dissection/filtration stages of analysis, and at the overall substantial similarity stage of analysis. Early on, expert testimony gained a greater role in look and feel cases merely as a result of the technical complexity and difficulty such cases usually present. As the infringement tests have moved beyond the simplicity of the Whelan rule and focused more on dissection and filtration,

the role of expert testimony has become, and will increasingly become, a mainstay of these cases. One can expect, for example, that expert testimony will be pivotal in arguing for or against application of "externalities" and other limiting doctrines in the filtration process, and in assessing the significance of similarities in the overall analysis of substantial similarity.

I. WHERE MUCH OF AN INTERFACE IS NOT PROTECTABLE, A HIGH DEGREE OF SIMILARITY IS REQUIRED

Several recent decisions have stated that where much or most of an interface is not protected or protectable, a high degree of similarity will be required for a finding of infringement. For example, the courts in the Apple v. Microsoft, Atari v. Oman, and Capcom v. Data East cases noted that "virtually identical" copying would be required in a case in which the allegedly similar elements are largely unprotectable. Thus, if the courts apply the newer limiting doctrines more in the future, the degree of similarity required for a finding of infringement may rise.

* * * * * * *

From the highest point of view, if there is any larger picture forming at this point in the evolution of the "look and feel" cases, it is a picture beginning to form around the abstraction/filtration/comparison test and its variants in the Circuits (sometimes under different terminology). The evolution continues, and with luck the picture will emerge more clearly.