January 1988

Expanding Copyrights in Software: The Struggle to Define Expression Begins

Paul R. Lamoree

Follow this and additional works at: http://digitalcommons.law.scu.edu/chtlj

Part of the Law Commons

Recommended Citation
Available at: http://digitalcommons.law.scu.edu/chtlj/vol4/iss1/3
EXPANDING COPYRIGHTS IN SOFTWARE:
THE STRUGGLE TO DEFINE
"EXPRESSION" BEGINS

Paul R. Lamoree*

I. INTRODUCTION

Computer software is most likely the variety of intellectual property least understood by laymen and judges. It is at the same time the variety for which the legal rules are least developed and the variety capable of having the greatest economic impact in the era of "competitiveness." Congress chose in the Computer Software Copyright Act of 1980, at the largely unexamined recommendation of an industry committee,¹ to effect protection and encouragement of software by very minor amendments to the Copyright Act.²

The copyright area itself is characterized by concepts and categories foreign both to ordinary common sense and to the language of the computer industry, and by a limited base of carefully elaborated judicial decisions.

It is the purpose of this article to offer a perspective on innovative judicial efforts to deal with software by relating certain aspects of programming to the corollary elements of literary creation. It argues that from this perspective, two recent and potentially seminal judicial decisions in cases of partial copying, one purporting to follow the other, are in fact wholly irreconcilable, and that the two, respectively, tend to arrive at altogether different outcomes for the struggle. It further observes that a third current decision poses a conflict in the circuits on the fundamental question of whether copyright can protect against anything more subtle than mechanical "line-by-line" copying of the entire program.

In particular, this article examines those decisions in light of an

* Paul R. Lamoree is a litigation partner with Watson, Ess, Marshall and Engass in Kansas City, Missouri. His substantive specialty is intellectual property. L.L.B. Yale Law School, 1967; B.S. Yale College, 1964. All rights reserved.


earlier effort to address the problems with an eye to predicting what aspects of a program can and should be protected against appropriation.

II. THE COPYRIGHT MODEL

Under the Act, one secures copyright protection for a program simply by writing it; preserves it merely by affixing a notice of copyright if it is "published;" and might register it by filing copies of a small fragment of the "source code listing" embodying the logic and mechanisms of the program with the Register of Copyrights. Representations of screen images may also be filed as audiovisual works or as literary compilations.

A certificate of copyright issues routinely for a computer program, just as it does for a book or photograph, without anyone having "read" the materials filed, much less having given consideration to what rights are being conferred.

There is no built-in process to guide the courts comparable to the administratively negotiated description of the allowed claims of letters patent. Even when disputes later arise and the copies of the "listing" are tendered as evidence of the content of the copyrighted work, the filings provide no more than obscure grist for expert testimony. No flow charts or other documentation explaining the program's use, purposes, or the means by which the cryptic source code implements the purpose need accompany the partial code sheets. Consequently, the question of what it is that the copyright holder owns is left for subsequent judicial determination.

Publicly sold versions of the program ordinarily contain only "object code" — a compilation of 0's and 1's which is readable only by the machine. What was originally written, called "source code," is rarely published. A subsequent author's determinations of what is proprietary to another must be made within the traditional copyright law categories of "copying," "adaptation," "access" and "substantial similarity," but without the ordinary benefit of being able to review the prior work in its entirety to extract the sense of the work in the medium in which it was originally authored. In most jurisdictions electing the copyright model for protection also leaves the determination of whether one program infringes the rights of the owner of another to the standard of an ordinary, reasonable person.

In traditional areas of copyright, such "reasonable persons" have risen to the occasion in determining whether "My Sweet Lord" infringes "Oh Happy Day," whether Captain Marvel comics infringe Superman comics, and whether the characters in the McDonaldland TV commercials infringe those in the H. R. Puf'n'Stuf children's show. The ear and eye may suffice in such contexts, although jurors know little of music theory or artists' techniques.

Whether two computer programs are substantially similar presents rather more subtle questions. Similarity in how they appear to work can be highly misleading, whether the appearance is judged from watching the screen displays and reviewing the printouts or by extrapolating from the verbal instructions in the user's manuals. For example, any spreadsheet program does pretty much the same functions as any other, although some have additional features. All produce columnar screen displays and printouts in accordance with a familiar format adopted from paper ledgers, and many have nearly identical keystrokes as commands to perform common functions. Even an experienced user can fail to discern which spreadsheet program is loaded.

Another consequence of adoption of the copyright model for software protection derives from a fundamental difference between the substantive law of copyright relative to that of patents. Copyright is derived from the common law protection against plagiarism, and accordingly focuses largely on the question of whether the alleged infringer in fact produced his work by "cribbing" from the copyrighted work.\textsuperscript{5} Whereas a patent may be infringed by one who arrives quite independently at the same mechanism on which a patent has issued. "[Copyright] law imposes no prohibition on those who, without copying, independently arrive at the precise combination of words or notes which have been copyrighted."\textsuperscript{6}

This distinctive focus of copyright has played a major role in decisions regarding software, and frames the issues for litigation in a copyright case very differently than in other areas of intellectual property.

Thus, Congress' adoption of the copyright model issued a call to battle for a decade of struggle between jurisprudence and the computer industry. The first guns have sounded.


III. THE SEMINAL DECISIONS ON SIMILARITY

Three decisions have delineated the issues in cases of appropriation of commercially valuable aspects of a program. Synercom Technology, Inc. v. University Computer Co.,\(^7\) Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.,\(^8\) and Broderbund Software, Inc. v. Unison World, Inc.,\(^9\) differ fundamentally on what aspects of a program are subject to copyright protection, on what evidence the trier of fact should inspect for similarity, and on the roles of expert and lay testimony. They differ implicitly on what the author should register for copyright in order to secure the broadest protection.

Recently, a fourth case, Plains Cotton Cooperative v. Goodpasture Computer Services,\(^10\) puts in doubt whether there can be any meaningful protection for such property.

A. Synercom: Order and Sequence as Unprotectable Idea

Synercom Technology arose before the Software Act and out of a technology less sophisticated than that associated with today's issues of cloned operating systems and Lotus look-alikes. Indeed, the copyrighted work in that case was not a computer program. However, even the relatively primitive technological context of that case posed inescapable tensions between the conceptual categories of software and those traditional to prior copyright law.

The plaintiff and defendant EDI\(^11\) had each developed a program in the FORTRAN\(^12\) language for doing engineering structural analysis on mainframe computers. The copyrights were in plaintiff's users' manuals and in nine separately registered format illustrations describing techniques for preparing punched cards to be fed into a computer running Synercom's STRAN\(^13\) program. The cards were to be punched in accordance with the pattern which Synercom had devised to make it easier to feed the user's data to its program. EDI's "Sacs II" program would have called for very different data.

---

8. Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986).
12. FORTRAN ("FORmula TRANsactions") is a widely used programming language especially suited for mathematical procedures and engineering applications.
13. STRAN ("STRucture ANalysis") program.
patterns, but EDI had interposed a preprocessor program in FORTRAN to reorient data punched to STRAN patterns into what Sacs II required.

EDI's preprocessor conversion was commercially vital. It allowed EDI to market Sacs II to the large base of users familiar with the use of STRAN. The STRAN pattern was by no means simple or self-evident; the court found that the plaintiff had invested $500,000 in development, including the training of customer personnel.

EDI effectively conceded at trial that its manual infringed Synercom's manuals. The importance of Synercom lies in its approach to the issues presented by plaintiff's claims that EDI's preprocessor program infringed Synercom's copyrights in the "formats." The court addressed those issues in terms that would have been equally applicable if the parties' works had been programs written in different computer languages.

Judge Higginbotham analogized broadly to the provisions of § 1(b) of the Copyright Act which gives the owner of a literary work the exclusive right "to translate the copyrighted work into another language or dialect," and volunteered an opinion that

[it] is as clear an infringement to translate a computer program from, for example, FORTRAN to ALGOL, as it is to translate a novel or play from English to French. In each case the substance of the expression (if one may speak in such contradictory language) is the same . . . with only the external manifestation of the expression changing.

Whatever the merits of that analogy, Synercom posed the


15. Id. at 1013, n.5.

16. The analogy may well be too facile. Whereas translating a work from one human language to another artfully is somewhat more than a matter of transliteration and adjusting for syntactical factors like word order, gender and case, the differences in the inherent logic of different high-level computer languages (e.g., as between "structured" languages like Pascal or a subroutine-centered language like COBOL) are so great as to suggest that the broader categories of "adaptation" and "derivative works," which encompass turning the novel into the play, provide a more suitable analogy than "translations." See Q-Co. Industries, Inc. v. Hoffman, 625 F. Supp. 608, 615-16 (S.D.N.Y. 1985), correctly observing that the test of whether a derivative work infringes, disregards differences between the works and focuses on what has been appropriated and its importance to the new work. See also Findings Nos. 4 and 7 in SAS Institute, Inc. v. S&H Computer Systems, 605 F. Supp. 816, 818 (M.D. Tenn. 1985), which would support registration of Release 79.5 of plaintiff's own program as "derivative" of its Release 76.2, although by one measure only 30% of the code was common to both. If unauthorized, such a derivative work infringes the prior work. See note 46, infra.

Converting an existing program to run under a different basic "operating system" control program ("MS-DOS," "Pick," "Unix," "VMS") and/or in a "hardware" environment of
fundamental analytical problem of applying the axioms of copyright in literary property to the subject matter of software. EDI argued that its preprocessor program was an original implementation of the idea or principle of Synercom's graphic formats (and manual texts), which copyright cannot protect, unrelated and essentially dissimilar to the expression of the idea which is protected.

This underlying but elusive distinction lies at the heart of the copyright model which has been extended to software. Whereas a patent grants a temporary monopoly over the novel conception revealed in the patent disclosure, a copyright “does not give a monopoly over an idea . . . but merely protects against the unlawful reproduction of an original work.” In conventional copyright terms, it is the artist's expression of the idea which may not be appropriated by another.

On the face of it, EDI's argument appears sound, if ever the distinction can allow wholly dissimilar programs directed at producing the same data output to coexist without infringement. Two programs intended to yield, e.g., a tabular display of the conventional general ledger balance figures of a corporation, would not ordinarily be presumed to be similar for purposes of copyright infringement merely because the data output was similar.

In fact, as explained below, the two parties' works did not yield comparable outputs. Additionally, inspection of the FORTRAN source or object code for EDI's preprocessor would have revealed no direct parallels to the Synercom writings; programming code does not include the kind of graphic shapes, visible arrangements or the kind of explanatory user directions that were in the manuals (except for trivial, non-functional literal constants for screen labels or prompts, like “Enter board dimensions:” and/or of the cursory text that might appear on a “help screen”).

distinctive architecture or capacities (single- vs. multi-user, distributed processing vs. central, memory-intensive vs. I/O-intensive) can present comparable challenges. See e.g., the description of the problems encountered in preparing a work-alike of a modest $49.95 children’s program written for Apple IIe's to run on IBM PC's, in Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1130-31 (N.D. Cal. 1986).

17. Patent law decisions give no basis for expecting patent protection for computer programs themselves, standing apart from industrial processes or the apparatus they may control. See In re Bradley, 650 F.2d 807 (C.C.P.A. 1979) (claim for means of controlling multi-process computer functions allowed, where hardware for implementing means also claimed); Diamond v. Diehr, 450 U.S. 175 (1981) (process patent for curing rubber not unpatentable because the process includes steps directed by computer program); Cf. Parker v. Flook, 437 U.S. 584 (1978), and Gottschalk v. Benson, 409 U.S. 63 (1972) (each declining to find invention where the novel feature of a process was in the controlling program).

18. 462 F. Supp. at 1012, n.4, citing and quoting Granite Music Corp. v. United Artists Corp., 532 F.2d 718, 720 (9th Cir. 1976).
Synercom contended that its idea was the abstract conception of a manner in which user data could be presented to the computation modules of its program and that the arrangement or order and sequence for the data constituted the expression of its authorship. Of necessity, it argued that its “expression” was less concrete than that part of its work which could have been reproduced into a visual medium (e.g., by photocopies, drawings by hand, video images). EDI indisputably had access to, and had appropriated, that order and sequence.

The argument has its merits. In copyright contexts, the medium is not the message. It is clear, for example, that a musical composition embodied in sheet music can be infringed by live or phonorecorded performance. The converse also applies. One who listens to a musical or dramatic performance and produces written parts or dialogue scripts from memory has “reproduced” it. (E.g., Leo Feist, Inc. v. Demarie.\(^{19}\)

Nonetheless, it would be conceptually difficult to entertain a contention that an Ansel Adams photograph of a Grand Canyon sunset infringed passages of a similar theme in Dvorak’s “From The New World,” or that “Pelias and Melisande” on canvas infringed a scene in the opera on the same theme. The order and sequence of elements in a work of authorship ordinarily may be part of the expression. However, infringement surely requires similar ordering or sequence of elements of like kind. The order, sequence and relative duration of the notes in the main coda from Beethoven’s 5th are each paralleled in the Morse code dit-dit-dit-da for the letter “V”, but the elements ordered in the two works, and the works themselves, are incomparable.

In Synercom, the elements of plaintiff’s graphic formats and of defendant’s preprocessor program were as different as those of, e.g., an oil painting of sunlight and a tone poem on the theme of darkness. Synercom’s illustrations instructed the eye and hand of a human being to take numbers from specifications and worksheets and to keypunch it in the STRAN sequence. EDI’s program was what 17 U.S.C. § 101 now defines as a “set of statements or instructions to be used directly or indirectly in a computer,” directing the machine to take numbers on cards punched in the STRAN sequence out of that order and to reorient them into the very distinct sequence required by the computational routines of Sacs II.

Moreover, there was no common ordering of the elements of

\(^{19}\) 16 F. Supp. 827, 828 (W.D. La. 1935).
Synercom's formats and EDI's code. Synercom's graphic elements were ordered to the STRAN sequence of punched holes, whereas EDI's verbal commands were ordered by the "syntactical" confines of FORTRAN in producing the other sequence.

Whatever rights of authorship might have subsisted in plaintiff's bare sequence, plaintiff manifestly had no intellectual or property interest in either the different sequence required by EDI's Sacs II or EDI's expression, in programming code, of means to alter data into the Sacs II order. From that perspective alone, the case should have been an eminently simple one.

On the other hand, there was evidence of plagiarism in copying the manuals' texts and in the appropriation of the commercial advantage of starting with the STRAN input formats. The court therefore attempted to enunciate a general rule within recognized categories of copyright protection for its denial of relief on the formats. It had to do so without the benefits of having in evidence a tangible embodiment of the allegedly infringing work, i.e., a printout of EDI's source code, or of having expert testimony on the dissimilarities in elements, function and technical rules governing the arrangement of (a) the command elements of EDI's work and (b) the data elements in Synercom's format.

The court found parallels to the formats in the recognized category of copyright protection of blank forms. That parallel has since figured in each subsequent software case. It observed that whereas the "sequencing of data for simplified access to the computer programs" was the utilitarian substance of the formats, Synercom's formats communicated it "by their placements of lines, shaded art and words." In reciting Copyright Circular 32's declaration that "there is no way to secure copyright protection for the idea or principle behind a blank form or similar work, or for any of the methods or systems involved in it," the court concluded that it was those artistic elements, not the substance, in which plaintiff had copyrights.

From those thoughts the court proposed a far-reaching analysis of dissimilar meanings for "idea" and "expression" in artistic works and functional works. According to the authorities, it said,

In cases of literary or artistic works, and works of a similar character, in which the form, arrangement or combination of ideas represents the product of labor and skilled effort separate

20. 462 F. Supp. at 1012.
21. Id. at 1014.
and apart from that entailed in the development of the intellectual conception involved . . . the medium of expression is entitled to protection by copyright against adoption by another in similar form, arrangement, and combination.\textsuperscript{23}

By contrast, the court said that among utilitarian works, such as computer programs, it may be that less can be owned:

If order and sequence is the expression, the skilled effort is not separable, for the form, arrangement, and combination is itself the intellectual conception involved. It would follow that only to the extent the expressions involve stylistic creativity \textit{above and beyond} the bare expression of sequence and arrangement, should they be protected. (emphasis added)\textsuperscript{24}

In so indicating, Judge Higginbotham accurately presented the central difficulty in software copyrights: in aesthetic works, the order and sequencing of the elements may constitute the “art” protected by the law; “‘Nevermore,’ quoth the raven,” or “A rose would smell as sweet by any other name,” can lose the savor of the originals, and concomitantly the legal protection. For a utilitarian program, there can be a number of techniques for producing each small result within the overall program such as determining the displaying of the date. Many of them are functionally equivalent in a gross sense.

Nonetheless, some solutions are elegant, and some pedestrian. Some are among the conventional tools of the trade, and some will reflect significant advances in the state of the art.

Which of them the programmer selects, or invents, and merges into the overall work, may be of little consequence and no visibility to the user, but his choices determine the precision, efficiency and speed with which the desired result is achieved.

Other choices restrict or increase the languages into which the program can be readily translated, and/or the kinds of computers on which it can run. These choices represent the programmer’s creative and skilled work no less than more visible stylistic flourishes.

Aesthetic elements are largely confined to aspects of the user interface, such as striking video characteristics in screen displays, mouse activation of icons\textsuperscript{25} in lieu of selection from menus or per-


\textsuperscript{24} \textit{Id.} at 1014.

\textsuperscript{25} “Icons” are screen images visually symbolizing functions. For example, a stylized wastebasket may signify discarding a file. The function is usually selected by moving a screen image of an arrow to the icon, using a “mouse.”
emptory status messages like "wait" instead of homey ones like "I'm thinking." These are largely incidental to the core capabilities of a program, but may be of great importance to the commercial success of some classes of mass-market software, and often entail a high proportion of the coding of such products. They can also be the aspects of such programs potential competitors most desire to clone.

User interface characteristics had no place in *Synercom.* STRAN and Sacs II were aimed at an engineering market, and arose in an era of punch devices, before the advent of interactive keyboards and screens, much less color graphics and voice synthesis. The idea-expression dichotomy there may have been exaggerated because the record presented no occasion for recognition of such aesthetic expression.

B. Whelan: *Order and Structure As Protected Expression*

What must be filed to register a program for copyright is a fraction of the programming code in its visible, source, embodiment. That requirement surely suggests that what is distinctive and protected by a Software Act copyright extends beyond user interface elements displayed when the program is run. The suggestion is strengthened by the circumstance that much software has no user interface. Many programs control machines to which the digital components are peripheral (machine tools, VCR's, engine fuel systems), or direct system-level computer functions that never manifest themselves to the user.

The *Synercom* definition of "expression" would, if extended to Software Act cases, deny copyright protection to the "arrangement, order and structure" in all computer-capable works. Such an extension was considered and rejected on carefully reasoned and artfully expressed grounds in a major recent decision, *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.* It was also rejected, on the authority of *Whelan,* in *Broderbund Software, Inc. v. Unison World,*

26. "Coding" is the detailed process of creating a program by writing sequences of verbal commands to govern each step to be done. The commands, in the vocabulary and syntax of a programming language, constitute the "source code." See Whelan Assoc. v. Jaslow Dental Laboratories, 797 F.2d at 1230-31 (3d Cir. 1986).


28. But see the discussion of *Broderbund Software,* infra, where the court constricted its consideration to the user interface, to the exclusion of the aspects of program addressing the machine and the data.

where, as will be noted, an enormously broader degree of protection was afforded on the basis of a seductively simple reading of some small part of Whelan.

The primary differences between these two decisions was dictated by the records the courts considered: In Whelan, expert testimony on similarities in the central routines of the programs was taken and given great weight; in Broderbund, the court limited its consideration to the sensible menu screens, and deemed itself bound under Ninth Circuit decisions to rely on application of an ordinary reasonable person standard of perceived similarity.

In Whelan, the situation presented precisely what had been hypothesized in Judge Higginbotham’s Note 5: the translation of a program from one computer language into another.

Both the copyrighted program and the allegedly infringing program served to keep books and guide administration of dental prosthetics laboratories; each performed the same functions for the user, each using much the same internal file structures to do so, and each generating similar data for the lab. In addition to their near-identity in external function, there were inescapable similarities in the visible stylistic aspects of the two programs when they executed, i.e., in the menus apprehensible to the user’s senses, and in the labeled screen outputs or data displays.

Those observations, however, were treated by the court as merely beginning the factual and legal inquiry. It cast the legal issue as being one of whether, in the absence of literal copying of the programming code, copyright could protect the basic structure, sequence, and organization of a copyrighted program. That phraseology was more than a paraphrase of Synercom, even though the earlier decision was clearly pertinent, because the elements ordered in the copyrighted work in Whelan were program instructions.

Literal copying of the registered code was clearly absent in the narrow sense, because with respect to source, the two programs were in different and dissimilar languages; and because, with respect

---

31. See the illustration that is Appendix A to the opinion, 797 F.2d at 1249.
32. Id. at 1224, 1248.
33. Footnote 1 to the passage provides a sharp cautionary note to those inclined to citation by headnote or by court synopsis: Although Whelan is already cited, as it is above, for the proposition that copyrights extend "beyond a program’s literal code to their structure, sequence and organization," footnote 1 declares that the opinion uses the three words interchangeably, as synonyms. The content of the single concept must be discerned in the context of the opinion.
to object code, they were for different machines. The copyrighted program had been written in 1979 in EDL, a language specific to an early microcomputer, the IBM System 1. The infringing program, however, was written four years later in BASIC, to run on IBM's new standard for microcomputers, the Personal Computer.

With narrow exceptions for use of disassemblers, there can be reproduction of program code only (a) by wholesale mechanical copying, on a computer, of the entirety of the object code distributed to user-licensees — which can be discerned almost as readily in that industry as it can in the record business or (b) where there has been an employment or business relationship between owner and infringer giving the latter access to the former's source code.

34. "Microcomputers" are computers based on "micro-processors," single-chip logic processors in which memory management functions, arithmetic logic units, storage registers, and related elements are integrated. They are often desk-top units.

35. Disassemblers, simple versions of which are part of the debugging utilities provided with many systems, blur the line between reverse engineering and outright copying. Running object code, the binary number sequences that the machine reads but which are meaningless to the human senses, through a disassembler produces human-readable assembly language commands, from which an expert programmer can discern all the logic and structure of the original program. This is so even if the original was compiled from source code written in a higher level language. From the assembly language, such a programmer can, albeit tediously, reconstruct the original flow chart of the program, and can proceed to recode it. For authority that doing so is legitimate, see E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485, 1490, 1492 (D.C. Minn. 1985). While the recoding may entail any number of equivalent variations from the first programmer's choices, including a different language, those variations can largely disappear when his reconstruction is recompiled, resulting in object code effectively identical to what was disassembled.

36. Software piracy cases often entail "copying the exact number and sequence of bytes or items by which Plaintiff's program causes the machine to operate." Apple Computer, Inc. v. Formula International, Inc., 562 F. Supp. 775, 782 (C.D. Cal. 1983), aff'd, 725 F.2d 521 (9th Cir. 1984). There it was noted that:

    Such appropriation is quite feasible; the testimony from experts offered on this motion [for a preliminary injunction] confirms that the process of duplicating or copying a program, once it is put into usable form in a ROM or diskette, is almost absurdly simple. Diskettes can be copied for a minimal cost by anyone with rudimentary technical skill and a few pieces of equipment, and ROMs require only slightly greater sophistication.

    Id. at 783.

    Any general-purpose computer can copy a program onto diskette or tape, with utilities built into the computer's operating system; indeed § 117 of the Copyright Act of 1980, as amended, expressly authorizes such copying for archival purposes (making a reserve copy of the program in the event of damage to the original). Reasonably priced tools are widely advertised and readily available for reproducing a program from and/or onto a ROM chip.

37. A case was posed in SAS Institute, Inc. v. S&H Computer Systems, 605 F. Supp. 816 (M.D. Tenn. 1985), where access to source was obtained by abusing a user license, and the offending work was a port to different equipment, effected by re-coding done by programmers given the licensed source. The result was wholesale, but not verbatim, reproduction. The court's findings, at 822-26, provide a reliable guide to the varieties of smoking gun similarities which can be developed in a well-prepared case; they range from near-identity in the
Source code is usually protected as a trade secret, by nondisclosure except subject to a confidentiality agreement, and — unlike other literary works — is ordinarily neither published when the program is sold on disk or tape nor displayed to the public.\(^\text{38}\) Source copying can be partial and subtle: particularly useful or innovative routines or modules can be appropriated into otherwise quite different code, and can become part of the appropriator's standard library of functions or tools.

One defendant in *Whelan*, Jaslow Laboratory, had engaged plaintiff's assignor to develop software to manage the operations of its prosthetics business, and had authorized the developer to market the resulting program to others on a royalty basis. Whelan was formed by the chief programmer of that name after delivery of the program, called Dentalab, and succeeded to the developer’s rights. Whelan appointed Jaslow its sales agent for it on a best efforts basis.

Dentalab was obsoleted for general market purposes by the passage of the System 1 and its EDL language. Jaslow's principal hired programmers to develop the allegedly infringing BASIC program, which he called Dentcom, to be marketed among labs acquiring the new PC's. He and a brother formed a company also called Dentcom to sell it, terminated the marketing contract with Whelan, and gave the latter notice that any sales of Dentalab would be actionable disclosure of Jaslow's trade secrets. When Whelan continued sales, Dentcom sued on that ground. Whelan counterclaimed for copyright violations, and removed the action (reversing the parties).

At trial, Whelan's expert testified\(^\text{39}\) from a review of the par-

code for 44 subroutines and 69 of 70 parallel interface routines, to incorporation of a module serving no purpose in the copy and ungrammatical and incoherent lines of code remaining after characters had been revised to disguise the origin. In this connection, but at a much higher level of technological sophistication in analysis, see E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485, 1492, 1494-97 (D.C. Minn. 1985). Another finding, at 818, establishes the nonreality of a literal identity requirement: plaintiff's own latest upgrade lacked verbatim identity with its predecessor in 67-70% of its lines of code.

38. *Whelan* was not unrepresentative in this particular situation because the programs were sold in uncompiled source, to be used with the interpreters in the purchasers' computers, but the court deemed the circumstance nonmaterial. 797 F.2d at 1231, n.20.

39. In one of its most consequential rulings, the court of appeals rejected, as potentially misleading and as of doubtful value in copyright cases "involving exceptionally difficult materials, like computer programs," the evidentiary rule of *Arnstein v. Porter*, 154 F.2d 464, 468-69 (2d Cir. 1946), *cert. denied*, 330 U.S. 851 (1947) previously adopted for the Third Circuit in copyright cases generally, in *Universal Athletic Sales Co. v. Salkeld*, 511 F.2d 904, 907, (3d Cir. 1975) *cert. denied*, 423 U.S. 863 (1975) whereby the intrinsic test of substantial similarity, i.e., whether what was appropriated is enough to make the challenged work infringing, had to be determined by the response of an ordinary observer; and expert testimony
ties' source codes that the BASIC Dentcom was not merely a direct translation from EDL of Dentalab, but that the file structures and screen outputs were virtually identical, and that the five subroutines basic to each of the programs performed almost identically. Jaslow's expert testified to "substantive differences in the programming style, in programming structure, in algorithms and data structures," but his report acknowledged "overall structural similarities" between the two.\footnote{Whelan, 797 F.2d at 1228.} Compare the "total concept and feel" test discussed in \textit{Broderbund} \footnote{Broderbund, 648 F. Supp. at 1137.} discussed \textit{infra}.

The trial court found all issues for Whelan. On copyright infringement, it found "that the Dentcom system, although written in another computer language from the Dentalab, and although not a direct transliteration of Dentalab, was substantially similar to Dentalab because its structure and overall organization were substantially similar."\footnote{Whelan, 609 F. Supp. 1307, 1321-22 (E.D. Pa. 1985), as paraphrased at 797 F.2d at 1229.}

On the legal question of what constituted the protectable expression, the trial court, per Judge Van Artsdalen, observed that other dental lab programs accomplished the same functions and incorporated many of the same ideas, but that

There are many ways the same data may be organized, assembled, held, retrieved and utilized by a computer . . . . The expression of the idea in a software computer program is the manner in which the program operates, controls and regulates the computer's receiving, assembling, calculating, retaining, correlating, and producing useful information either on a screen, printout or by audio information.\footnote{609 F. Supp. at 1320. Fairly read, the "screen, printout or . . . audio" phrase has application only to the subset of interactive, user-oriented programs which have display and reporting functions. However, the inclusion of calculating and data-flow functions places the focus correctly on how any program operates, not just on how it appears to a user, if at all. Two programs developed from a common flow chart may be implemented quite differently by a variety of programmer (and/or compiler) options regarding, e.g., what arithmetic logic unit operations shall be involved in what order, what memory variables shall be sustained, and what information shall be held in volatile memory rather than written off to disk for later recovery.}

Judge Van Artsdalen may have viewed the desired result for the program, in the sense of the application for which it was written, as the idea or principle of the work. The passage above also states:

\begin{quote}
could be admitted only on the extrinsic test of substantial similarity, \textit{i.e.}, whether the alleged infringer in fact made use of the copyrighted work. 797 F.2d at 1232-33.
\end{quote}
Different computer systems may functionally serve similar purposes without being copies of each other. There is evidence in the record that there are other software programs for the business management of dental laboratories. There is no contention that any of them infringe although they may incorporate many of the same ideas and functions.44

There is a vast conceptual middle ground between "idea" in this macrocosmic sense, and in the sense of the means selected by the author for implementation of a particular program function or set of functions. Copyright cases have yet to address whether a novel algorithm, or a novel application of a known algorithm, is merely an idea, or instead an expression of the idea that a particular function be implemented.45

On Jaslow’s appeal, the Third Circuit, in a remarkable opinion by Judge Edward Becker, acknowledged it confronted a “case of first impression in the courts of appeals.”46 It affirmed, upon reaching the conclusions (a) that the National Commission on New Technological Use of Copyrighted Works (CONTU), and, by exten-

44. Id.; The plaintiff in Apple Computer, Inc. v. Formula International, Inc., 562 F. Supp. 775, 782 (C.D. Cal. 1983), 725 F.2d 521, 523-25 (1984), “introduced evidence, which was not directly controverted by Defendant, that numerous methods exist for writing the programs involved here that would be 98% compatible with Apple computers, yet not so similar to Plaintiff’s particular sequence as to infringe its copyright.” That case involved fundamental operating system programs, where mechanical limitations may restrict some programming options; most applications programs allow more flexibility in seeking out equivalents.

45. CONTU evidently intended a contention that a given task could only be accomplished by specific programming code to be only an affirmative defense to infringement of that segment of code, rather than a ground for re-defining the whole of the first program to include that code as an uncopyrightable manifestation of a principle of logic, inadvertently placed in the public domain at publication. It stated, at p. 20 of its Final Report, that “when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement.” See E.F. Johnson Co. v. Uniden Corp. of America, 623 F. Supp. 1485, 1492 n.4, 1501 n.16 (D.C. Minn. 1985), treating this as equivalent at the bottom line to a finding of no substantial similarity.

The CONTU approach, although procedurally appealing, is too narrow in its focus on verbatim identity. Even a novel algorithm may be implemented by any number of different programming instructions. That difficulty was obscured in E.F. Johnson by the circumstances (a) that the programs in evidence were in assembly and hexadecimal machine notation, written in rudimentary instructions and inflexible syntax, and (b) that infringing routines had been copied so slavishly as to use instructions inappropriate to defendant’s hardware and to repeat plaintiff’s mistakes. Id. at 1494-96.

Indeed, the court there found only one code number in plaintiff’s program essential to the application. Id. at 1494. It also found other elements infringed notwithstanding dissimilarities apparent in a line-by-line comparison of the parties’ disassembled coding (it deemed the dissimilarities unconvincing, in view of evidence those elements had in fact been adapted from a disassembled copy of Johnson’s program). Id. at 1497-98.

46. Whelan, 797 F.2d 1222, 1224
sion, the Congress which adopted its recommendations as the Software Act without generating a legislative history of deliberations of its own, "intended copyright protection to extend beyond the literal code," and (b) that "the detailed structure of the Dentalab program is part of the expression, not the idea, of that program.”

Judge Becker noted that the CONTU report had included flow charts, with source and object code, as "works of authorship in which copyright subsists," and that the circumstance "demonstrated that the Commission intended copyright protection to extend beyond the literal code," but that CONTU had looked to caselaw development for resolution of the "dichotomy between idea and expression.”

The reference to flow charts was highly suggestive. A flow chart schematically defines, in standardized graphic figures, each of the principal operations of the program to be written, and something of the logical relationship among those gross operations. Because a flow chart is devised before coding begins (although often after the data structures are settled), and because it does not detail the specific means of effecting the operations it conceptualizes, one could readily view it as a diagram of the problems addressed or of the idea for addressing them, rather than as an expression of the author's specific solution. Even Synercom treated the level of specificity characteristic of a flow chart as "requiring substantial imagination, creativity, independent thought and exercise of discretion,” but as being so far from a completed program in terms of abstraction as to make chart and code "really two different expressions of the same idea, rather than two different versions of the same expression.”

It is certainly true that the flow chart is remote in content and kind from the finished program. Programs in different languages, for different computer environments, can be developed from a single flow chart without sharing a single line of code, and/or without sharing anything by way of front end appearance or output format. The flow chart is akin to a scenario or story outline, and is as separately copyrightable from various adaptations or projections of it as are a stage play, a libretto, a photoplay, or a novel. That observa-

47. Id. at 1241.
48. Id.
tion begs the question of infringement, however, as much in the computer context as it does in the literary. The photoplay (and the film made of it), although eligible for separate copyrights as a derivative work, may well infringe the scenario from which it derives. And the stage play may infringe both.\(^{51}\) If the preliminary flow chart were the proper material of software copyright in the relevant sense of a work from which a program is derived (not just in the sense that a rudimentary textual or pictorial work is protected against facsimile reproduction) — and it is routine in industry to impress copyright bugs or proprietary notices on flow charts no less than on shop drawings — then cases like Whelan might be decided without resort to the implementing code. Further legal developments can be expected on that thorny subject, especially in light of the ruling on analogous design specifications in Plains Cotton Co-op v. Goodpasture, discussed below.

The opinion recognized that the question was complicated by the distinction between literary or artistic works and utilitarian works, which had been noted in Synercom. It therefore began its analysis with *Baker v. Selden*, \(^{52}\) "a seminal case in the law of copyright generally," and "particularly relevant here because, like the instant case, it involved a utilitarian work, rather than an artistic or fictional one."\(^{53}\)

*Baker v. Selden* involved a book of blank forms and explanatory text for a new, simplified system of bookkeeping. Defendant Baker promoted a competing accounting system that reproduced Selden's forms, and Selden sued. The system was concededly not copyrightable (although dictum suggested it might be a patentable invention), and Baker's text was non-infringing. The issue was whether the forms embodied Selden's idea, or were part of its expression. The Supreme Court held that where the art, meaning Sel-

---

den's accounting system, "cannot be used without employing the methods and diagrams used to illustrate the book," or "similar" forms, the "methods and diagrams" are part of the conception, and are no more protected by copyright than the system in the abstract.\(^{54}\)

To Judge Becker, *Baker v. Selden* implied that: "[T]he 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." [emphasis in original]\(^{55}\)

He cited *Apple Computer, Inc. v. Formula International, Inc.*,\(^{56}\) where the functions a computer program performed were deemed a non-protectable idea, but the code expressing it by specific programming commands and techniques was protected from reproduction through reverse engineering.

The court found analogies in the decisions denying copyright protection to scenes *a faire*, "incidents, characters or settings which are as a practical matter indispensable . . . in the treatment of a given topic,"\(^{57}\) and in those limiting protection to "fact-intensive works" like news articles, historical works, and compilations.\(^{58}\)

In both areas, the court found the rationale to be "that there are only a limited number of ways to express factual material, and therefore the purpose of the literary work — telling a truthful story — can be accomplished only by employing one of a limited number of devices. [Citation] Those devices therefore belong to the idea, not the expression. . . ."\(^{59}\)

At that point the court shifted to consider the pragmatic implications of the possible alternative rules. The court discussed the rule granting protection to distinctive aspects of a program, to an extent comparable to that for fact-intensive works, beyond protection limited to literal copying, but not extending to anything necessary to effecting the function it serves by other specific means. It stated that

[W]e . . . believe that the rule would advance the basic purpose underlying the idea/expression distinction, 'the preservation of

\(^{54}\) Whelan, 797 F.2d 1222, 1236.
\(^{55}\) Apple Computer, 562 F. Supp. at 783.
\(^{57}\) See *Miller v. Universal City Studios, Inc.*, 650 F.2d 1365, 1372 (5th Cir. 1981).
\(^{58}\) Whelan, 797 F.2d at 1237.
\(^{59}\) Id.
the balance between competition and protection reflected in the patent and copyright laws..."\(^{60}\)

In so asserting, the court referred to its earlier citations for the proposition that coding consumes only about 20% of the effort in software development, and stated:

Among the more significant costs in computer programming are those attributable to developing the structure and logic of the program. The rule proposed here... would provide the proper incentive for programmers by protecting their most valuable efforts, while not giving them a stranglehold over the development of new computer devices that accomplish the same end.\(^{61}\)

It rejected as irrelevant in other copyright contexts the objections (a) that "computer programs are so intricate, each step so dependent on all of the other steps, that they are almost impossible to copy except literally, and that anyone who attempts to copy the structure of a program without copying its literal elements must expend a tremendous amount of effort and creativity,"\(^{62}\) (b) that the concept of program structure is too vague a guide for litigation,\(^{63}\) and (c) that progress in software is achieved on the backs of those who have gone before, and "requires plagiarizing in some manner the underlying copyrighted work."\(^{64}\)

Having determined that software cases required no more severe a standard than the conventional copyright test of substantial similarity, the Whelan court considered expert evidence of similarities in two external (user-sensible) areas and in one internal program structure and logic.

---

\(^{60}\) *Id.*, citing *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 466 F.2d 738, 742 (9th Cir. 1971).

\(^{61}\) *See Note, Copyrighted Computer Programs*, 68 Minn. L. Rev. 1264, 1290 (1984) ("almost the same amount of time as the original programmer expended").


\(^{63}\) Note, 68 Minn. L. Rev., n.55, at 1292, *quoted in Whelan*, 797 F.2d at 1237-38.

IV. EXTERNAL MANIFESTATIONS OF STRUCTURAL SIMILARITY

A. File Structures

Whelan complained that Dentcom had reproduced the file structures of Dentalab. Jaslow did not dispute the fact, but argued that (a) file structures, like blank forms, are merely part of the idea, under Baker v. Selden, and (b) protecting file structures would particularly raise the problem of foreclosing an entire area of programming art, in that its expert witness had testified that there were few, if any, alternatives.

File structures are classified here as external because they are largely, if not fully, discernible from screen templates guiding data entry, in that the prompts calling for certain information to be keyed and the space provided for each response define what types of data are stored, whether each item is textual, numeric, or mixed, and much of the items' inter-relationships. The nature of file structures is discussed hereinafter.

The court rejected that reading of Baker, on the basis of Third Circuit decisions construing its placement of forms on the idea/expression spectrum less mechanically; and held that, compared to the Selden forms in terms of comprehensiveness and complexity, the file structures of the programs were "sufficiently informative to deserve copyright protection." It approached more charily the question of whether Dentalab's structures must nonetheless be denied protection as the only framework in which a program for that function could be realized, so the idea was merged in the expression and that enforcing copyrights in them would be tantamount to granting a monopoly over programs of the type. It based its conclusion on the record before it that other programs for the function "might use significantly different file structures," and declined to reverse on the basis of a matter of technical expertise on which the trial court had rejected contrary opinion testimony as unconvincing. However, it suggested it might have reached a different conclusion "had the defendants offered more evidence to support their position" on the point, and noted, in dictum of which more will no doubt be heard, that:

It is true that for certain tasks there are only a very limited number of file structures available, and in such cases the structures might not be copyrightable and similarity of file structures

65. Whelan, 797 F.2d at 1243.
66. Id.
67. Id. at 1243, n.43.
might not be strongly probative of similarity of the program[s] as a whole. We are simply not convinced that this is such a case.68

The author strongly suggests that the dictum reserves judgment too cautiously. In the first place, defining "file structures," in the narrow sense of ordering the elements of an electronic blank form, is a preliminary and relatively trivial part of program design, and would be recognized as such even by lawyers who build tickler files in dBASE III or use Pascal's "Record" command to define simple sequential files.

In the second place, it would be difficult, for example, to identify a software submarket, in terms of size and type of machines or size and type of business, which does not include a large number of plain or fancy business accounting packages to track cash receipts and expenditures, receivables and payables, payroll and associated taxes, and the like, each incorporating certain file structures dictated by accounting conventions.69 In such fields, the software industry and users of its products would be disruptively surprised by a rule leaving it a litigable issue whether all other packages infringe copyrights in the first (or in Selden's forms). The conventional function restricts the tolerable range of variation in fundamental file structures to such a point as to make protection a socially undesirable restriction on hitherto lively competition.

There is another example broached in the opinion in a slightly different context: it is difficult to visualize how any key word in context full-text retrieval system, whether for office text files, litigation documents or a public database, could be devised without reliance on indexing schemes fundamentally similar to those at least externally familiar to users of "Westlaw" and "Lexis."70 It would be radical indeed if a court were to conclude that such an indexing

68. E.g., expenditures posted to a ledger payables detail file by date; amount in decimal numbers; and account-type code, reserving space for date, amount and item of payment; those items carried from there in running totals for encompassing categories to an income statement report, and to either current or fixed assets-and-liabilities files, as dictated by the account-type code.

69. Computer file "indexes" are the second logical tier of any file structures more advanced than those used for private electronic phone lists or baseball card collection data.

70. The same tabular data may be stored and accessed according to any of several different logical schemes, which can require very different computer operations to effect its manipulation and retrieval. An indexed sequential scheme relies on computer-generated tables of the physical locations of the data items, conceptualized as a sequence of identically-structured index cards or pages, each item identified by the "distance" from a beginning point at which it begins. Network and hierarchical schemes place the items along the branches of an "inverted tree structure," rather than along a line of only one dimension. Relational schemes array data items along multiple parallel lines associating the items on each line according to recurring "key" items.
scheme was the exclusive property of the first to use it in some broad or narrow context.

If file structures are to be considered for aspects of the expression, development of the law in this area may require consideration of refinements on the blank form analogy for file structures to distinguish indexed sequential, network, hierarchical and relational structures.\[71\] Certainly the selection of one such type of structure is in the realm of idea, but the programmer's choice has inescapable consequences on the choices of (1) what file-placement data (keys) shall go in the separate records, tables, relations, etc., and (2) what logic will be adopted in the operative routines of the program to effect storage and retrieval.

Those factors pose issues of whether, e.g., two payables files differentiated only by placement data fields or columns included to implement the respective file-integration schemes are substantially similar; of whether coding to gather data from a network file structure is really dissimilar from coding to gather it from a relational structure; and of whether the answers differ on the basis of whether the program was written \textit{ab initio} or within a database management system environment where the DBMS\[72\] supplies that which differs between the two sets of structures. These are not productive lines of inquiry. \textit{Synercom} showed greater wisdom with respect to the blank forms analogy.

\subsection*{B. Screen Outputs}

The second aspect of similarity considered was in screen outputs. The trial court had stated with disconcerting informality that:

One thing is apparent to even the most casual untrained observation of the operation of the two systems — the visual screens that are displayed are almost identical in format and even in use of abbreviations and terminology. There is credible evidence that prospective users and customers at trade shows found no substantial difference between the [two], and considered them to be the same.\[73\]

On appeal, Dentcom argued that the screen's appearances were

\begin{itemize}
\item A Database Management System (DBMS) is a fundamental computer management program permitting items of user-defined data to be entered, stored, manipulated and retrieved selectively.
\item Whelan, 609 F. Supp. at 1322.
\item Aspects of Dentalab were the subject of four registrations, two covering the original and enhanced code, and two on installation and user's manuals. All were "literary" on Form TX. The trial court had found all four valid. 609 F. Supp. 1312, 1323.
\end{itemize}
(or should have been) covered by a variety of copyrights other than those on the registrations of which Whelan claimed, and that the screen outputs “bear no relation to the programs that produce them.”

The opinion acknowledged that “screen outputs are considered audio-visual works under the copyright code,” and that Whelan “asserts no claim of ... infringement with respect to the screen outputs,” but held that “the only conclusion to be drawn from the fact of the different copyrights is that the screen output cannot be direct evidence of copyright infringement [of the registered literary work].” It declared the trial court’s quoted statement to be a valid one of evidence, under Fed. R. Evid. §§ 401 and 403:

There is no reason, however, why material falling under one copyright category could not be indirect, inferential evidence of the nature of material covered by another copyright.

The court nonetheless acknowledged, on a point that would be overlooked with significant consequences in Broderbund that there was force to the argument that

It is still possible ... that the risk of unfair prejudice would outweigh the probative value of the evidence ... one might argue that, because the screen outputs are visible and easily understood (at least as compared with the obscure details of computer programs), they might have disproportionate influence on the trier of fact [although] they might say very little about the under-

74. Whelan, 797 F.2d at 1244.
75. Id. at 1244, citing Williams Electronics v. Artic International, Inc., 685 F.2d 870, 874 (3rd Cir. 1982) (video displays in computer game protected as audio-visual works, 17 U.S.C. § 102(a)(6)); Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 749 (N.D. Ill. 1983) (same, but noting that operational aspects of “PAC-MAN” game program, also registered as literary work, were infringed although screen display was not reproduced); and Stern Electronics, Inc. v. Kaufman, 669 F.2d 852, 855 (2d Cir. 1982) (literary registration of code proper to protect program yielding audio-visual displays, but suggesting the Form TX registration would not protect against a different set of instructions producing visually identical displays). See also, Atari, Inc. v. North American Phillips Consumer Electronics Corp., 672 F.2d 607 (7th Cir. 1982), where the only copyrights claimed in the “PAC-MAN” video game were audio-visual (no doubt because of the legal uncertainties as to whether any but the sensible, artistic elements of a game in any medium are eligible for copyright). Compare Chamberlin v. Ursis Sales Corp., 150 F.2d 512 (2d Cir. 1945) with 1 Nimmer § 2.18(H)(3). Infringement of very narrowly construed rights in the “size, shape, and manner of movement,” pie-wedge “mouth” and gobbling sounds of the plaintiff’s hunter and hunted figures, and in its “regeneration” process, were held infringed.
76. Whelan, 797 F.2d at 1244.
77. Id.
78. Id.
79. Id. at 1245.
lying program.80

On the record before it, the court of appeals found the Rule 403 argument unpersuasive of reversible error, noting it was no matter of plain error because defendant had not objected to the testimony on screen appearances (which, from what was said below, was indirect), and that the party against whom screen outputs were asserted "could easily explain their limited probative value."81

*Whelan* could be sanguine about litigants' ability to overcome the disproportionate influence of screen appearance on ordinary observers because it had accepted the expert testimony required to overcome it on intrinsic similarity. In jurisdictions less liberal on the evidentiary question, the ruling on screen outputs may prove pernicious.

Screen similarities might better be considered as possible misrepresentations of origin or performance under § 43(a) of the Lanham (Trademark) Act,82 where customer response surveys have an accustomed and directly relevant role; or as evidence of deceptive trade dress.

C. *The Five Subroutines*

Having held that it would not apply the ordinary observer test in copyright cases "involving exceptionally difficult materials, like computer programs,"83 and that literal identity between the parties' works was no more needed for infringement of software than of other works,84 the Third Circuit made its most important contribution. It proceeded to valuate expert testimony of "whether the most significant steps of the program are similar."85 The testimony of plaintiff's expert had focused on "the programs that had the primary . . . tasks of the system, and also ones that manipulate files . . . programs which actually showed the flow of information, through the system . . . ."86

The opinion described "the crucial issue in this case [as] the similarities and differences in the programs' *structures,*"87 and reviewed the adequacy of the evidence in terms of testimony on five central subroutines for "invoicing," "day's end," "month-end,"

80. *Id.*
81. 797 F.2d 1003 (N.D. Tex. 1983).
83. 797 F.2d at 1232-33.
84. *Id.* at 1245-46.
85. *Id.* at 1246 (emphasis added).
86. *Id.* at 1246.
87. *Id.* at 1247 (emphasis in original).
“accounts receivable” and “order entry.” The testimony had been as to various of these that “they are doing pretty much the same thing, in the same fields in the same files, and accomplishing pretty much the same results;”\(^8\) “[t]he same choices are offered . . .,” on the screen, to “change, skip, cancel;” they “do the same thing in the same order” (except for “flipping the order” in which two reports are printed); and that “because the same file structure and the same program steps are followed, the same overall flow takes place in both systems.”\(^9\)

One may certainly quibble at the weight given testimony that both programs used a flag, a character placed in a separate logic field, to identify to later routines which orders had been invoiced,\(^9\) given that setting a flag in a field or column of a record or table (or in a memory variable) is a nearly unavoidable journeyman tool of the programming trade. It was also of debatable wisdom to find common work of skill or art from testimony that both programs produced, in not quite the same order, month-end reports for “product group,” “customer sales analysis,” “accounts receivable aging,” “AR reports that had to do with service charges,” “age file balance” and “total accounts receivable.”\(^9\) Few manual or automated accounting systems could be found which differ from that pattern.

Nonetheless, \textit{Whelan} imposed a pragmatic and progressive framework on issues of the validity and infringement of software copyrights, within which factual questions presented by programs of any nature can be treated; and in which the defining balance of economic interests on which the law impinges can be coherently addressed.

V. \textit{BRODERBUND V. UNISON WORLD: THE USER INTERFACE AND SIMILARITY IN MENUS AND SEQUENCES OF MENUS}

Within 60 days of the Third Circuit decision, the quotable holding of \textit{Whelan} with respect to the protectability of the structure, sequence and organization of the logical elements of Form TX literary copyrights, of which screen output may be indirect evidence, was broadly extended to make similarity in “the structure, sequencing and arrangement of screens” registered as “audio-vis-

\(^{88}\) Id. at 1247.
\(^{89}\) Id. at 1247-48.
\(^{90}\) Id. at 1247.
\(^{91}\) Id. at 1248.
ual" works effectively conclusive evidence of infringement.92

At the same time, Whelan's tests of whether file structures were so fundamentally dictated by the functions to be performed by the copyrighted program, and of whether commonly-structured files were similarly manipulated in the primary subroutines of the respective programs, were combined and then altered into one of whether "there is no other conceivable way to structure" the "menu screens and sequence of screens" in a program based on same "underlying ideas."93

Broderbund, like Synercom and Whelan, fits the plagiarism model, in that it arose out of the introduction of a competitive software product by a party given full access to the object code of the infringed product in the course of a terminated venture between the two authors. Even more than in Whelan, and in contrast to the other cases involving attempts to appropriate readily-available, mass-market products by mechanical copying (the Apple and "PAC-MAN" cases), parallels between the programs were shown to be due to the conscious, even admitted, translation from one language and environment to others hypothesized in Synercom's dictum, but in a context of more problematic reverse engineering.

The formative distinctions of Broderbund, however, were procedural. First, the plaintiff had wisely learned from Stern Electronics94 and Midway Manufacturing,95 to register its code and screens separately, respectively as literary and audio-visual works; and had sued under both, and for trademark infringement and unfair competition as well. The reported decision revolved only a separate trial of the severed issue of liability on the audio-visual copyrights.96 Second, the court's reluctant exclusion of technical testimony, discussed below, had a perhaps equal impact on the decision.

The registered work was embodied in a program called The Print Shop, developed to exploit the graphics capabilities of Apple home computers to allow users to create customized greeting cards,

---

93. Id. at 1132.
94. Stern Electronics, Inc. v. Kaufman, 669 F.2d 852 (2nd Cir. 1982).
posters, banners and signs, using bit-mapped type faces and graphic elements otherwise unavailable.97

Broderbund, exclusive licensee for distribution of the program, had retained Unison World to attempt to port it to run on other microcomputers, after Broderbund itself had failed to produce a version for the IBM Personal Computer. Unison bargained for rights to an IBM-compatible version, but negotiations failed after several weeks of attempts to recreate the typefaces of the program for the very different IBM graphics and PC-DOS environment.

Unison continued its efforts to clone The Print Shop. After ten months, it brought out its own PC version, which included several new capabilities, and which, for all the opinion shows, had not either a single line of code source in common with the predecessor, or — except as may be implied by what follows — a single segment of the instructions organized as plaintiff had organized its own.

However, when run by the user, Unison's Printmaster produced top-end menu screens, ten user interface screens in two function areas ("Greeting Card" and "Sign"), and additional screen interface elements for the "Picture Editor" function, all corresponding in appearance to those produced by plaintiff's program.98

The degree of similarity in the screens was intentional. The evidence was that, when the competitive phase of the project began, Hughes, who had been assigned to the project by Unison during the negotiations for rights to reproduce plaintiff's program, had reused the screens he developed toward a perfect reproduction. He had never, however, had access to plaintiff's source code. Unison had only been given diskette copies of the program in object code, and his assignment had been to reverse engineer it, i.e., to independently devise means for generating the same appearance on the different hardware.

The opinion does not indicate that Hughes disassembled plaintiff's program to do so. There is no technical reason why one would in order to generate comparable screens; the logic and flow of a program to create menus and other screen displays is routine and familiar to any applications programmer. There are indications that Hughes' screen programming effort had focused on the much more sophisticated problem of reproducing plaintiff's proportional type faces and video characteristics on a dissimilar system — on which plaintiff's work provided no aid.

97. The product had been an exceptional commercial success, selling 500,000 copies at $49.95. Id. at 1130.
98. Id. at 1131, 1135.
Moreover, the court felt disabled to consider expert testimony of such a matter. Although it expressed a preference for the Whelan approach of integrating analytic dissection by experts with lay evaluation in making findings on similarity,99 it was bound by the Ninth Circuit's adoption, in Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp.,100 of the Arnstein v. Porter rule101 that the response of an ordinary reasonable person is the test of intrinsic infringement.102

Because only rights in the audiovisual aspects of the work were tried, the idea/expression dichotomy was approached quite narrowly. Consideration was limited to superficial aspects of the user interface discernible to the eye when the program ran. Defendant thus had to argue that the idea which merged with the expression, to which protection was to be denied under 17 U.S.C. § 102(b) and the authorities previously cited, was the external appearance of what was displayed to the user.

Defendant . . . argues that the idea underlying the menu screens, input formats, and sequencing or screens in "Print Shop" is indistinguishable from its expression. Any menu-driven computer program that allows its users to print greeting cards, signs, banners, and posters will have a user interface substantially similar to that in "Print Shop," defendant contends, because there is no other conceivable way to structure such a program.103

The court rejected that sweeping contention by reference to evidence of the Stickybear Printer program from a third party, which had substantially the same functions, and thus was found to express the same underlying ideas:

Yet the expressions of those ideas are very different. The menus, screens, and sequence of screens in the two programs are different. The entire structure and organization of the user interfaces are different.104

Within the cramped scope of the court's consideration, that conclusion was unavoidable.

Defendant's fallback position emphasized another copyright distinction, which will be made critical if consideration of audiovi-

99. Id. at 1136.
100. Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp., 562 F.2d 1157, 1164-65 (9th Cir. 1977).
103. Id. at 1132.
104. Id.
suay copyrights in software may be severed from the underlying literal program elements, as was attempted here. While lesser aspects of utilitarian literary works are regarded as expressive, rather than artistic, literary works, and that circumstance led to the focus in *Synercom* and *Whelan* on organization and sequence, the statutory rule for pictorial and graphic components of utilitarian works is that

... [T]he design of a useful article shall be considered a pictorial, graphic or sculptural work if, and only to the extent that such design incorporates pictorial, graphic or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article.

That defense argument did not prevail on the record in *Broderbund*, because Unison had reproduced the screens as nearly as was feasible, and infringement could be found in purely stylistic factors such as relative size of typeface entries, selection of non-functional parallels in phraseology (e.g., “Choose a Font,” rather than “Select . . . .”), and arbitrary function names (“Screen Magic”). In fact, functional matters had been overlooked in the rush to market: the strongest indication of direct copying was that, on one screen of the IBM program, the user was still prompted to press “Return,” although the key performing that function is labeled “Enter” on the PC keyboard.

It is somewhat difficult to reconcile the court’s foregoing constricted statement of what was being protected (which would seem to allow appropriation of all of the user interface of a popular program but the colors, letter sizes, precise prompt phrasing, and module names) with other passages in the decision (a) seeming to adopt the much looser “total concept and feel” of similarity enunciated in *Sid & Marty Krofft*, and *Roth Greeting Cards v. United Card*.

---

105. The very narrow range of protection afforded in *Atari, Inc.*, 672 F.2d 607, made its manner of giving separate consideration to audiovisual rights much less consequential to expectations in the industry. That court used standards of comparison for infringement developed in the pictorial and audiovisual fields, and therefore finally based its finding of infringement only on appropriation of the most distinctive, almost trademark-like, aspects of the central game characters, while noting that anything more than slight differences in appearance and movement would belie the requisite degree of similarity. 672 F. Supp. at 618-19. Compare the treatment of infringement of the literary copyrights in the same game in *Midway, Inc.*, *supra*, note 75, where substantial similarity was treated as something short of absolute identity in the object code, and was found on the basis of evidence of identity of between 89% and 97% of the 128,000 units of code (bits in the object code). 564 F. Supp. at 753-754.


108. 562 F.2d 1157, 1167 (9th Cir. 1977).
and (b) including factors that were procedural, and only remotely aesthetic, among findings of infringement:

The ordinary observer could hardly avoid being struck by eerie resemblance between the screens of the two programs. In general, the sequence of screens and the choices presented...; and the method of feedback to the user are all substantially similar.

... [T]he structures of the "Main Menu" screens; the "staggered" layout of 3-2-3-2-3, totalling thirteen graphics; the "tiled" layout [offset overlaying of submenus] of $5 \times 7$ in both programs; [T]he fact that the "tiled" option disappears... in medium-sized graphics mode; the use of only left and right arrow keys...; the offering... of only three types of lines...; and the fact that both programs require the user to create the front of the printed product before creating the inside of it.110

If the decision is to be taken as establishing that "the overall structure, sequencing and arrangement of screens" in the user interface of a program are protected by copyright, or — broader yet — that a test of the "look and feel" to an ordinary observer is applicable to software cases generally, the profession must anticipate a good deal of wrestling with the question of what features of a user interface can be "identified separately from, and are capable of existing independently of, the utilitarian aspects" of the program.

The sequence in which menus (or command line instructions, or icons) appear and present their options; the presence or disappearance of some options when others are selected; the number of capabilities offered (e.g., kinds of typefaces that can be incorporated in output); the sequence in which modules can be invoked (e.g., inside design only after cover design); and the method of feedback to the user may well reflect fundamental factors in the logic and structure of a program, determinative of the utilitarian functions it performs, which reach far beyond the merely pictorial or graphic.111

Such factors are undoubtedly reflected in a flow chart of the functions and procedures of a program, which comes into existence before screen display coding even commences, and which is itself explicitly a subject for literary copyrights.

If infringement of audiovisual rights can be found from such factors, without expert-guided examination of its source and/or ob-

110. 648 F. Supp. at 1137 (emphasis added).
ject code, or of its operative logical structure, little remains of Whelan's holding that sensible, external similarities in "screen output cannot be direct evidence of . . . infringement" of literary copyrights in the program itself (and thus, at most, are rebuttable),112 or of Atari's restrictive reading of the protective scope of only audio-visual copyrights;113 or of Midway's114 and Stern's115 recognition that altogether unrelated program instructions can yield identical screen appearances.

Thus, the true impact of Broderbund as precedent may not be settled for some time. A flurry of litigation against lookalike programs comparable to that which followed the reversal in Apple v. Franklin116 is to be expected until the questions are settled.

V. POSTSCRIPT ON PLAINS COTTON v. GOODPASTURE: FIFTH CIRCUIT TO FACE ISSUE OF ORGANIZATIONAL COPYING

In a brief but pregnant January 21, 1987 opinion affirming denial of a preliminary injunction against distribution or sale of a program for receiving commodity prices and for transmitting and accounting for commodity trades, allegedly copied from the plaintiff's such program, the Fifth Circuit has tentatively rejected Whelan's "structure and organization" approach and opted for a potentially very consequential extrapolation from Synercom.

Plains Cotton Cooperative Ass'n, v Goodpasture Computer Services, Inc.,117 is explicitly a procedurally limited decision, sustaining rulings below, on a premature record, with respect to the preliminary injunction standards of likelihood of success on the merits and irreparable harm. It comes in a case, however, which could, if and as it progresses, present for decision a number of the unresolved issues identified herein.

Quite apart from the rulings on general software copyright law discussed below, and although nothing in the opinion indicates awareness of the distinctiveness of its subject matter, Plains Cotton is the first appellate decision regarding copyrights in a class of property that is rapidly gaining economic importance. We can expect it to generate further litigation and, probably, a distinctive line of precedent addressed to its special problems.

112. Whelan, 797 F.2d at 1244.
113. See note 98.
116. See notes 40, 47, 58.
Plains' program operates an on-line public database on the owner's mainframe computer, made available to subscribers for remote access from their terminals, akin to the Quotron stock market quotations services and various other real-time videotext systems, but also incorporates an interactive transaction processing system for accepting and accounting for purchase and/or sales orders from the subscribers. A system with many of these features is now being made available on a government subsidized basis to telephone subscribers in France, and is the subject of frantic scrambling for position by television cable companies and the Bell Operating Companies in this country.

As it stands, the Plains Cotton decision constitutes an express election not to follow Whelan on a partial record in which the copyright owner had argued that parallels in the "organizational structure" of the subject programs sufficed to establish a likelihood of success on the merits of infringement even if the trial court chose, as it did, to accept the testimony of the defendants' expert on extrinsic similarity (use of access) that there had been no "verbatim, line-by-line copying." Instead, the court of appeals chose, at least while "one step removed from the actual merits," to adopt Synercom's conclusion

118. Public databases are fee-per-inquiry dial-up services by which a computer user may obtain information on any covered subject, from an index of periodicals to a table of the ductility of alloys.


120. 807 F.2d at 1262.
that "organization and configuration of information fed to the computer . . . were not protected by copyright," and to extend that rule very materially by holding, where there was evidence that "many of the similarities" between the programs were "dictated by the externalities of the cotton market" in which the programs were used:

The record supports the inference that market factors play a significant role in determining the sequence and organization of cotton marketing software, and we decline to hold that those patterns cannot constitute 'ideas' in a computer context.

The double negative in the quoted ruling may reflect reservation of final judgment on that specific holding, but it does leave as tentative conclusions (a) that aspects of a program's organization dictated or determined to a significant extent by market factors are non-copyrightable, and (b) that a showing of "the reproduction of the organizational structure of a software system, outlined generally in the software's design specifications" results in "similarity . . . on a level not protected by . . . copyright."*

A. Screen Formats Dictated by Market Factors As Uncopyrightable Ideas

The court explained what it meant on point (a) by example: Footnote 4 notes "[f]or example, appellees' witnesses testified that their cotton marketing program was designed to present the same information as is contained on a cotton recap sheet, within the confines imposed by use of a computer," whereas plaintiff's expert was unfamiliar with the cotton marketing industry and thus had no opinion as to "whether the similarity" between the programs arises from an attempt to convey the same standardized information to the user.

From the quoted testimony, it is evident that the sequence and organization which the opinion indicates to have been dictated by the market was merely the organization of the format in which transactional data was presented to the user on the screen in one part of the user interface (both parties had sought to recreate the appearance of a familiar paper form). Conversely it was not organization of the logical, operational program structure, in the sense of

---

121. Id. at 1262 (quoting Synercom Tech. v. University Computing Co., 462 F. Supp. 1003, 1014 (N.D. Tex. 1978)).
122. 807 F.2d at 1262.
123. Id. at 1260.
124. Id. at 1262, n.4.
the most significant steps of the program, as Midway Manufacturing was paraphrased in Whelan; or of the "primary . . . tasks of the system . . . which manipulate files . . . [and] actually showed the flow of information, through the system," as in Whelan's holding on the five subroutines. 125

What the court in Plains Cotton was rejecting from Whelan seems to have been those simplest parts of it which most squarely took issue with Synercom: those relating to "data structures" (which the Third Circuit had, with reason, compared to Synercom's data input formats); and those regarding "screen outputs" (which came to be the distortingly narrow focus of the opinion in Broderbund).

B. Functional and Design Specifications As A Level Not Protected By Copyright

On its conclusion (b), above, the Fifth Circuit in Plains Cotton confronted a record on which it found:

[Defendants-appellees'] GEMS is very similar to [plaintiff-appellant's] Telcot on the functional specifications, programming, and documentation levels. In fact, several pages of the GEMS design manual appear to be direct copies of pages from the design manual appellees created at CXS. The main difference between the two systems is that Telcot is designed to work on a mainframe computer, whereas GEMS is designed for a personal computer. 126

CXS had been a joint venturer with Plains Cotton: the four programmer defendants had gone to CXS from Plains Cotton's employment when a departing officer of plaintiff had formed CXS and caused it to enter into a venture with plaintiff to develop a microcomputer version of the Telcot mainframe system. During the venture period, with access to a tape of the Telcot source code, they had produced a document setting out design specifications for a micro version, but had done no programming (coding) for it. They then moved on to Goodpasture, taking along a copy of the Telcot programming designs (by which the court is understood to have meant the source code brought from Plains Cotton to CXS). Within 20 days of their arrival, they had completed a design of a personal computer version. Some seven or eight months later, Goodpasture went to market with a rough, incomplete program in-

125. Whelan, 797 F.2d at 1246.
126. Plains Cotton, 807 F.2d at 1258.
corporating that design, although the software was not fully operational by the date of the injunction hearing.

Defendants admitted literal copying of Telcot only to the limited extent of one subroutine, but that had been replaced and the responsible programmer discharged before suit, and that appropriation therefore was not part of the request for preliminary relief. Otherwise, with a possible exception for the screen aping the "Cotton Recap Sheet" in each program, direct copying was not at issue on the truncated record.

Instead, the partial record came close to presenting, for the first time, the question of whether flow chart elements, singly or in the aggregate, can be infringed without literal copying of the registrant's implementing code.

On the record before it, the circuit court clearly felt there was credible evidence to support findings that "appellant's programs were too large to have been copied and modified in the . . . time it took to create GEMS;" that the mainframe Telcot could have been altered to run on PC's only with changes so "enormous" that "rewriting the programs would have been faster than modifying them;" and that similarities at the design level were such as to have resulted plausibly from the programmers' reliance on "just experience and industry knowledge." It held, with respect to corollary issues posed by the count for misappropriation of trade secrets (in the form of the business procedures embodied in the program's functioning), that the terminating employee is entitled to apply in new employment "the experience, knowledge, memory, and skill, which he . . . gained while there employed."

Thus, it viewed the case before it as one in which only the basic organization of Plains Cotton's program was seriously contended to have been appropriated; one in which it had not been shown that anything of that organization was other than what was necessary for any cotton exchange program; and one in which relief could not be afforded without depriving the defendant programmers and the public of the benefit of the very idea of such a program.

The record appears not to have clearly defined what the func-

---

127. But see the discussions, in connection with Whelan and Broderbund, supra, of whether similar screen appearances ought to imply similar works of literary authorship.
128. A post-trial record in the case could come closer yet.
129. 807 F.2d at 1260.
130. Id. at 1261.
131. Id. at 1261.
tional and/or design specifications in question contained, or in what detail their contents were set forth; the opinion certainly does not do so. From the limited allusions to them, design specifications appear to have been much more than a RFP (Request for Proposal) for software — a list of desired functions and capabilities ("functional specifications"): the four programmer defendants had spent a year at CXS to develop the document of design specifications. Since they had arrived with the complete source for the mainframe Telcot they had already written for plaintiff, it must be assumed the CXS document involved some serious and detailed design specific to the contemplated PC version.

Indeed, their work may have involved the distinctly nontrivial design of distributed database method used to collect, sort, and pass around a network of microcomputers, i.e., the transactional data that Telcot handled in a very different on-line, multi-user mainframe environment.

Thus, although the cursory, procedurally-oriented opinion does not make the matter clear, it is probable that defendants took from the joint venture something akin to a completed flow chart for a micro cotton exchange program. It is quite clear, on the other hand, that plaintiff did not purport to own, or to have registered, anything regarding the venture’s works: Plains Cotton had an option to acquire rights in the CXS product, but was found not to have exercised it.

What was registered, after suit began on other claims, was evidently the source for the mainframe version, and, possibly, the documentation and specifications for it, to which the court refers as “the system.” Thus, Plains Cotton results in a decision on a full trial record, or if the question comes before the Supreme Court on certiorari as an apparent conflict between the Third and Fifth Circuits (and perhaps the Ninth), a decision could be forced on what rights inhere in flow charts (or design specifications), as well as on whether Synercom or Whelan is to be followed.

133. “Design specifications” and “design manual,” although commonly used in programming, are not terms of art to which a very specific content attaches. In this sense, they are quite unlike “flow chart.” In general usage, such specifications could be expected to include descriptions of data structures and the relationships among different data structures (including keys and index definitions), copies of formats for report printouts, and, to extents varying among jobs, layouts for screen outputs, input screens, and principal menus. They ordinarily reflect a lesser level of specificity and development than flow charts per se. In modest jobs, such specifications may suffice, so that the flow chart phase can be omitted.

134. Plains Cotton Cooperative, 807 F.2d at 1258.
135. Id.
136. Id. at 1259.
VI. Conclusion

Recent decisions in the Third and Ninth Circuits have produced greatly enhanced protection for software, and significantly more sureness as to the rules of lawful competition. Nonetheless, the tendencies in the two circuits are proceeding toward an unavoidable collision in two areas: (1) the role and admissibility of expert testimony in infringement cases; and (2) the degree to which literary copyrights may be overwhelmed by audiovisual.

The Plains Cotton decision puts the Fifth Circuit at odds with the Third and Ninth Circuits to a degree that may ultimately require Supreme Court resolution.

These areas of impending collision pose serious tactical quandaries for intellectual property practitioners and the software industry. Undoubtedly, audiovisual registration should be made for any program with a significant user interface, even where what might otherwise be considered artistic elements play a much smaller role in market appeal than is the case among the video games, screen graphics, and computer-assisted design products which previously were the constituency for audiovisual copyrights. Literary registration remains requisite, but a question has been raised as to whether it extends or restricts the graphic rights.

When infringement is to be asserted, decisions of real practical importance must be confronted as to whether to bring a single action on audiovisual and literary copyrights; whether severance of the two types of claims will benefit one party or the other; and whether expert testimony, if admissible in the jurisdiction (on the literary rights only), may not prove disadvantageous to the plaintiff who tenders it by negating the ordinary observer's impression of superficial similarities. Before Plains Cotton the substantive law seemed to have been clarified in many respects. Nonetheless, new areas of ambiguity had been brought forward. A computer program is undeniably more utilitarian than any other subject of copyright. It therefore poses a uniquely serious tension between the idea/expression poles which distinguish the content of copyright and patent grants.

The distinctive aspects of a program's realization of its underlying idea, defined in terms of its organization of instructions — in compiled, machine readable form (independent of source language or writing style) — have been held protected expression with respect to literary rights in it, over the objection that what is necessary to accomplish a pragmatic purpose cannot be monopolized under copyright.
The quasiTrademark graphic characteristics identifying specific software to users are protectable expression with respect to audiovisual rights in its appearance when displayed on a video monitor, just as Superman's spit-curl was protected in print media. Major conceptual and practical difficulties in the idea/expression categories remain to be resolved, however, where the aesthetic elements of the program as displayed are integrated with how it functions, i.e., where the structure of the visual elements correspond non-incidentally with the structure of the underlying program's organization of the functions it performs, and the means by which it performs them. The difficulty is greater where the aesthetic elements arguably derive from patterns in the public domain.

Some serious substantive problems remain unaddressed. Although one program can be found to be a mere translation of another into a different language under the Whelan test of basic organization (perhaps aided by Midway's consideration of comparisons of the bits and bytes in the object code), gross dissimilarities in the appearance of the source notwithstanding, object code comparison can be rendered meaningless by a port of the program to another computer architecture (for which the compiled object code may take on a very different appearance). The cases provide no clear standards either (a) by which to test whether one program that looks like another when disassembled, or when abstracted to a flow chart, but which was coded independently, with no more than user access to the other, infringes, or (b) on whether disassembled assembly language or flow charts are admissible.

Finally, there is still no case authority on whether a novel solution to a problem within the flow of a program, i.e., a novel algorithm, which cannot be confidently said to rise to the dignity of a patentable idea, can be protected as an element of the programmer's expression; or whether, instead, a solution of questionable invention and/or one not patentable as a practical matter because of the limited hospitality of the Patent Office to software applications, must go wholly unprotected. The highest art of authorship, and significant commercial advantage, may inhere in such a solution, without its occupying any large proportion of the code, and without its affecting much of the remainder of the program's structure.