IDEA: The Journal of Law & Technology
1994

A LIGHT IN THE FOREST OF COPYRIGHT PROTECTION AND COMPUTER PROGRAMS?

Andrew N. Parfomak [n.a]

(The Second Circuit Addresses The Applicability of Copyright Protection And Computer Programming--Computer Associates International Inc. v. Altai Inc., 23 U.S.P.Q.2d 1241 (2d Cir.1992).)

Decided June 22, 1992, Computer Associates International Inc. v. Altai Inc. [n.1], the Court of Appeals for the Second Circuit takes a bold step in the right direction, namely, in scrutinously examining where and to what extent copyright law has overgrown its proper bounds and expanded into unintended areas, and cutting back the dense undergrowth that has tangled the path of the most wary traveler in seeking to determine the metes and bounds of what extent of protection is provided to computer programs.

At Trial:

Computer Associates International Inc. v. Altai Inc.(I) [n.2]

The plaintiff Computer Associates International Inc., ("CA") brought an action against Altai Inc. ("Altai") for causes of action including copyright infringement and misappropriation of trade secrets, alleging that portions of CA's computer job scheduling program titled "CA-SCHEDULER" were infringed by Altai's similarly functioning programs titled "ZEKE", "ZACK" and "ZEBB." CA based its claim on its belief that portions of a critical module of the CA- SCHEDULER program *56 named "ADAPTER", a program responsible for providing an interface between the operation of the scheduling program which was normally utilized by the user, and the particular input/output requirements of the operating system and computer upon which the CA-SCHEDULER program was run. The beneficial feature of the use of the ADAPTER program lay in the fact that CA could provide to its customers versions of the CA-SCHEDULER program which would appear substantially identical to the user of the program, regardless of the type of computer and operating system the program was running on, as the ADAPTER program not visible to the user of the program would provide the appropriate intermediate layer between the program and the machine. The use of ADAPTER enhanced the marketability of the program, and the use of such an intermediate layer program allowed for CA to provide versions of CA-SCHEDULER for

different computers with a minimization of reprogramming, requiring only that necessary portions of ADAPTER be rewritten to accommodate the particular requirements of the host computer.

In the latter part of 1983, Altai whose president since 1988 was one James P. Williams and who had left CA in 1980 where he was a product manager, approached a long-standing friend and programmer at CA, one Claude Arney III to offer a position at Altai who was in the employ of CA for the prior five years. Williams intended to employ Arney, an experienced programmer, at Altai for the purpose of writing new versions of Altai's interface program, titled "OSCAR" which formed a portion of Altai's computer job scheduling program named "ZEKE" for a different computer operating system, IBM's MVS operating system for which ZEKE was then unavailable. In terms of intended function and operation, Altai's ZEKE and its interface portion OSCAR were analogous to CA's CA- SCHEDULER and its interface portion ADAPTER. Unbeknownst to Williams, when Arney left CA in 1984 to join Altai, he had taken copies of CA's ADAPTER program with him in contravention of his employment agreement with CA.

At Altai, Williams assigned Arney with the task of producing a new version of OSCAR which was to be the interface program between IBM's MVS operating system and Altai's ZEKE program. Arney, unknown to Williams or others at Altai, produced a new version of OSCAR, called "OSCAR 3.4" utilizing approximately 30% of the program code of ADAPTER which he had taken from CA. OSCAR 3.4 was utilized by Altai in the ZEKE program, as well as two others from 1985 to the second half of 1988 when CA filed suit alleging copyright infringement and misappropriation of trade secrets against Altai. Upon this notice, Williams confronted Arney, who then admitted his use of approximately *57 30% of ADAPTER's program code in OSCAR 3.4. Subsequently, on the advice of counsel, Williams had Arney identify the program code portions which had been taken from ADAPTER, and then to the exclusion of Arney's further participation had the copied portions of code removed and rewritten by other Altai programmers who were provided only with sufficient instructions directed at what the program code was to accomplish, but who were denied access to either Arney or the original program code of OSCAR 3.4. This newly rewritten version, named "OSCAR 3.5", did not contain any of the code of the ADAPTER program and was sent to all users as a "free upgrade" for the ZEKE, ZACK and ZEBB products and was included in all further releases of these products.

Suitwas filed on grounds of copyright infringement and misappropriation of trade secrets was brought to trial in the Federal District Court of the Eastern District of New York, before Judge Pratt sitting by designation. The ruling noted that the issue of misappropriation of trade secrets was not considered as under the facts of the case, wherein CA plead and proved facts that one tortious act, that of copying of the program code of the ADAPTER program was the basis of both the copyright infringement claim and the misappropriation of trade secret law claim and under 17 U.S.C. § 301(a) [n.3] the federal copyright issue preempted the trade secret misappropriation claim which arose under state law. [n.4] The trade secret issue being preempted, the remaining issue to be

decided by the court was "... whether Altai's OSCAR 3.5 infringes CA's ADAPTER." [n.5]

In satisfying the requisite grounds that copyright infringement had occurred which required a showing of ownership and copying of the copyrighted work, the Court first found that although the portion of the CA-SCHEDULER program which was in dispute, namely the ADAPTER program was not the subject of a separate copyright, it was nonetheless encompassed in the copyright registration of the CA-*58 SCHEDULER program and the subject of a valid copyright. [n.6]

Turning to the second ground, that of actual copying which required a showing of access to the work by an alleged infringer, and substantial similarity between the copyrighted work and the allegedly infringing material, the Court noted that Altai had conceded that there was infringement with respect to OSCAR 3.4, the Court entered into a lengthy discussion as to its finding that the OSCAR 3.5 program did not infringe CA's ADAPTER program. [n.7] As to the aspect of "access", the Court determined that although there may have been access to the ADAPTER code through Williams, the good faith effort to rewrite the offending portions of the OSCAR 3.4 code to produce a clean version, the OSCAR 3.5 program was satisfied. [n.8] Turning then to the issue of "substantial similarity" the Court first stated that Whelan Associates v. Jaslow Dental Laboratory [n.9] proposed a "bright-line" test [n.10], [n.11] for separating what was protectable expression and what was a protectable idea. This "bright-line" test was fatally flawed as computer programs typically comprised plural sub-programs and sub-sub-programs some of which might be copyrightable, but many of which were standard routines in the computer arts and uncopyrightable. The focus upon the "structure, sequence, and organization" analysis in Whelan failed to recognize that the seeming method or process by which the program operated could have little relationship to the written structure of the code, source or object code, as "... the two views of a computer program, as text and as behavior, are 'quite distinct".' [n.12] (The Court, declined to address the issue of the propriety of copyright protection for computer programs in light of 17 U.S.C. § 102(b)'s *59 prohibition against the applicability of copyright for "... process, system, method of operation ... regardless of the form in which it is described, explained, illustrated, or embodied in such work.") [n.13] Instead, the Court decided that the application of the abstractions test originally proposed in Nichols v. Universal Pictures [n.14] by Judge Learned Hand would be more appropriate, proceeding "in order of 'increasing generality' from object code, source code, to parameter lists, to services required, to general outline."

Moving to the analysis, infringement under OSCAR 3.4 being conceded, the Court turned to the question whether OSCAR 3.5 was substantially similar to ADAPTER. Applying the abstractions test and in great part relying upon the testimony of the Court's appointed expert, Dr. Randall Davis, [n.15] inquiry proceeded from the most concrete form of the program through increasingly more general aspects. At the most concrete level, that of similarity of the individual lines of program code of OSCAR 3.5 and ADAPTER, none was found as the OSCAR 3.5 program was purposely rewritten to remove any copied program code. At the next higher level of generality, that of the

structure and operations of macros and parameter lists, no similarity was found as the Court deemed that any similarity was dictated by the functionality of the program, part of which was based on information regarding the IBM operating system with which OSCAR 3.5 was to work with and in the public domain. No support to this claim was supplied by CA, and no "substantial similarity" was found at this level to support a claim of copyright infringement. At the next higher level of generality, that of the list of services provided by the program to its users, such was deemed to be dictated by the requirement of functionality of the program, and not to copying of the ADAPTER program's similar elements.

The opinion reproduced and seemingly adopted a quantified view of the various factors and their relative importance for evaluating the levels of similarity:

Code	1,000
Parameter Lists	100
Macros	100
List of Services	1
Organizational Chart	Nil [n.16]

*60 By this analysis in determining "substantial similarity", the most important factor, that of actual copying was nonexistent as the code had been rewritten, and CA had failed to carry its burden of proof with regard to the other factors. Accordingly, no copyright infringement was found by Altai of CA's ADAPTER program. [n.17]

The Court disposed of the trade secret issue as preempted by the copyright issue and dismissed the claim, [n.18] and after a lengthy discussion regarding the damages due for infringement of the ADAPTER program by the prior OSCAR 3.4 program, found damages in the amount of $364,444.

On Appeal:

Computer Associates International Inc. v. Altai Inc.(II) [n.19]

The decision was appealed to the Court of Appeals for the Second Circuit where it was reviewed by Judges Altimari, Mahoney and Walker of which Judge Walker wrote the opinion. [n.20]

At the outset, the Circuit Court noted its full support of the earlier decision and opinion of Judge Pratt and began its discussion with certain statements regarding computer program design. These statements began with the definition of "computer program" under the Copyright Act, [n.21] and then noted that in program design the first step was that of determining the program's ultimate purpose, after which its functions could be broken down into subroutines or modules for performing specific tasks within the overall program. Transfer of information between these subroutines required that they have similar "parameter lists" [n.22] and that the interrelationships between the subroutines could be considered to constitute the "structure" of the program. Only after the specification of these prior elements is the program "coded" first into source code, then object code. [n.23]

The Circuit Court reviewed the facts of the case at trial and then turned to the issue raised upon appeal by Computer Associates International, Inc. namely that "... the test applied by the district court failed to account sufficiently for a computer program's non-literal *61 elements. Second ... the district court erroneously concluded that its state law trade secret claims had been preempted by the federal copyright act...." [n.24]

Turning to the first appealed issue, the Circuit Court restated its understanding that:

As a general matter, and to varying degrees, copyright protection extends beyond a literary work's strictly textual form to its non-literal components. As we have said "[i]t is of course essential to any protection of literary property ... that the right cannot be limited literally to the text, else a plagiarist would escape by immaterial variations." This black letter proposition is a springboard for our discussion. [n.25]

CA's remaining claim was that, notwithstanding the rewriting of OSCAR 3.4 to OSCAR 3.5 in order to remove program code admittedly copied from CA's ADAPTER program, the resulting OSCAR 3.5 program nonetheless had a structure substantially similar to the ADAPTER program. [n.26]

Turning to the case law [n.27] dealing with the subject which established that non-literal structures of computer programs may also be protected by copyright, the Court recognized this as the rule of applicable law, but went on to state " H owever, that conclusion does not end our analysis. We must determine the scope of copyright protection that extends to a computer program's non-literal structure." [n.28]

Prior to the initiation of its analysis, the Circuit Court entered the caveat that it was not their intent that the present decision should control infringement actions "... regarding categorically distinct works, such as certain type of screen displays." [n.29]

The Circuit Court first noted that "copyright does not protect an idea, but only the expression of the idea" and that the copyright laws *62 appeared to be intended to "protect computer programs only 'to the extent that they incorporate authorship in programmer's expression of original ideas, as distinguished from the ideas themselves.' " [n.30] However, at the same time the "... essentially utilitarian nature of a computer program further complicates the task of distilling its idea from its expression" emphasis

added and that as such, computer programs "hover even more closely to the elusive boundary line described in § 102(b)." [n.31] Recognition of the necessarily utilitarian nature of computer programs begins the inquiry of the copyrightable nature of computer programming with Baker v. Selden, wherein the Supreme Court decided that the necessary forms incidental to the practice of Selden's method of bookkeeping described in his text were not properly the subject of copyright protection, and that the accounting text of Selden (providing instruction in his method of accounting) and a computer program were roughly analogous under 17 U.S.C. § 101. As Selden's ledger sheets were necessarily incidental to his method of accounting, similarly uncopyrightable are "those elements of a computer program that are necessarily incidental to its function." [n.32] However, the Circuit Court noted that although Baker v. Selden provided a good analytical foundation in the area of computer programs, the most developed system of analysis was proposed in Whelan v. Jaslow. [n.33] The broad view discussed in Whelan separated the "... purpose or function of the utilitarian work as the work's idea, and everything that is not necessary to the purpose or function as part of the expression of the idea ..." where the Whelan court had decided that the noncopyrightable "idea" was the broad concept of "the efficient management of a dental laboratory." [n.34]

The Circuit Court, affirming the reasoning of Judge Pratt, noted that the reasoning and decision in Whelan was deficient as failing to account for the differences in a program's static nature (as may be considered its written code) and the same program's dynamic nature (as when the written code functions on a computer), as well as the non-differentiating use of the terms "structure, sequence and organization" used in describing *63 the computer program in Whelan; Whelan had become too dated in view of the advance of computer science. [n.35]

The Circuit Court promulgated a new, more practical three-part test to be used by the District Courts to ascertain the necessary "substantial similarity" in order to maintain a claim of copyright infringement of a computer program.

> [A] court would first break down the allegedly infringed program into its constituent structural parts. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable material. Left with a kernel, or possibly kernels, or creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program. The result of this comparison will determine whether the protectable elements of the programs at issue are substantially similar so as to warrant a finding of infringement. [n.36]

In the elucidation of its test, the first step, "Abstraction", requires a court to:

> Initially, in a manner that resembles reverse engineering on a theoretical plane, a court should dissect the allegedly copied program's structure and isolate each level of abstraction contained within it. This process begins with the code and ends with an articulation of the program's ultimate function. Along the way, it is necessary essentially to retrace and map each of the designer's steps--in the opposite order in which they were taken during the program's creation. [n.37]

In the second step, "Filtration", the inquiring court in seeking to find substantially similar elements between two programs being examined is required to:

[E]xamine the structural components at each level of abstraction to determine whether their particular inclusion at that level was "idea" or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain and hence is non-protectable expression. [n.38]

The purpose of the analysis being to "define the scope of plaintiff's copyright." [n.39]

*64 In Circuit Court elaborated what it understood to be "considerations of efficiency and necessarily incidental", "external factors" and those "from the public domain."

Considerations of "efficiency" recognized practical considerations in the computer software industry, in that the focus of a great deal of a programmer's effort is directed in providing software in which the needs of the user are satisfied in the most efficient manner. Meeting efficiency requirements however may "so narrow the practical range of choice as to make only one or two forms of expression workable options", and thus remove the program's element from copyright protection as it may have merged with the underlying idea of the program and thus become uncopyrightable subject matter. [n.40] To make this determination,

[T]he court must inquire "whether the use of this particular set of modules is necessary efficiently to implement that part of the program's process" being implemented.... If the answer is yes, then the expression represented by the programmer's choice of a specific module or group of modules has merged with their underlying idea and is unprotected. [n.41]

Recognizing further that "efficiency is an industry-wide goal" and that there may be only a "limited number of efficient implementations for a given task" the Circuit Court noted both that it was possible that independent groups of programmers might contain the "same efficient structure" and that such might be equally evidential of independent creation as well as being evidential of copying; thus "... since evidence of similarly efficient structure is not particularly probative of copying, it should be disregarded in the overall substantial similarity analysis." [n.42]

Considering elements which would be "necessarily incidental", the Circuit Court recognized that programmers work within the constricted boundaries of the environment of computers and computer programs, and that their design choices are proportionately limited:

This is a result of the fact that a programmer's freedom of design choice is often circumscribed by extrinsic considerations such as (1) the mechanical specifications of the computer on which a particular program is intended to run; (2) compatibility requirements of other programs with which a program *65 is designated to operate in conjunction; (3) computer manufacturer's design standards; (4) demands of the industry

being serviced; and (5) widely accepted programming practices within the computer industry. [n.43]

The Court then approvingly noted that prior cases had already excluded from copyright protection for elements which were dictated by external requirements of a particular market for which the program was written, [n.44] for the type of on-screen navigation which was dictated by the type of computer hardware upon which the program was run, [n.45] and particular modules in a teleprompter program which would be considered inherent in any teleprompting program. [n.46]

Considering elements which would be from the "public domain," an inquiring court would need to remove from copyright protection "expression that is, if not standard, then commonplace in the computer software industry." [n.47]

As the third step of the required analysis, "Comparison", requires that those elements which have remained and had not been filtered out in the first "Abstraction" step and the second "Filtration" step would constitute what remained as work which may be of copyrightable value. [n.48] Only at this point would a comparison of the elements of the defendant's allegedly infringing program and the elements of the plaintiff's program remaining after the first two steps are made, as well as an assessment of the "... copied portion's relative importance with respect to the plaintiff's overall program." [n.49]

The Circuit Court went on to justify its new test by noting that it was in concert with the expressed objectives of and constitutional policies underlying the copyright laws, and in part to dispel the objections raised by CA and various amici who argued that such a test as proposed *66 would be a disincentive to programmers who improve programs due to the lack of broad copyright protection. [n.50] The Court noted approvingly that in Feist Publications, Inc. v. Rural Telephone Service Co., Inc. [n.51] the Supreme Court had dispelled the "sweat of the brow" or the "industrious compilation" theories which had been used to justify copyright protection upon compilations, noting that the intent of the copyright laws was not to reward effort, but originality. [n.52] From this case then, the Circuit Court stated that:

> [w]hile the Feist decision deals primarily with the copyrightability of purely factual compilations, its underlying tenents apply to much of the work involved in computer programming. Feist teaches that substantial effort alone cannot confer copyright status on an otherwise uncopyrightable work. As we have discussed, despite the fact that significant labor and expense often goes into computer program flow-charting and debugging, that process does not always result in inherently protectable expression. Thus Feist implicitly undercuts the Whelan rationale, "which allow[ed] copyright protection beyond the literal computer code ... [in order to] provide the proper incentive for programmers by protecting their most valuable efforts [citations omitted]." We note that Whelan was decided prior to Feist when the "sweat of the brow" doctrine still had vitality. In view of the Supreme Court's recent holding, however, we must reject the legal basis of CA's disincentive argument. [n.53]

*67 The Circuit Court further criticized the holding in Whelan v. Jaslow as being too expansive in its scope of protection [n.54], but noted that with *68 respect to the proper form of intellectual property protection which should be afforded to computer programs, that:

[T]he exact contours of copyright protection for non-literal program structure ... [were] not completely clear. We trust that as future cases are decided, those limits will become better defined. Indeed, it may well be that the Copyright Act serves as a relatively weak barrier against public access to the theoretical interstices behind a program's source and object codes. This results from the hybrid nature of a computer program, which, while it is literary expression, is also a highly functional, utilitarian component in the large process of computing.

Generally we think that copyright registration--with its indiscriminating availability--is not ideally suited to deal with the highly dynamic technology of computer science. Thus far, many of the decisions in this area reflect the court's attempt to fit the proverbial square peg in a round hole. [n.55]

Notwithstanding this apparent shortcoming in the law, the Circuit Court stated that its new test was believed to comport with the fundamental tenents of copyright doctrine, notwithstanding its expected effect in narrowing the scope of protection afforded to computer programs. [n.56]

The Circuit Court then turned to review the record at the lower trial, upholding the use of the expert, Dr. Davis who was instrumental in finding no substantial similaritybetween the OSCAR 3.5 and ADAPTER programs. [n.57] Turning next to the analysis of the evidence, the single point of criticism regarded Judge Pratt's critical analysis of OSCAR 3.5 and not the ADAPTER program.

We think that our approach--i.e., filtering out the unprotected aspects of an allegedly infringed program and then comparing the end product to the structure of the suspect program--is preferable, and therefore we believe that district courts should proceed in this manner in future cases. Furthermore, by focusing the analysis on the infringing rather than on the infringed material, a court may mistakenly place too little emphasis on a quantitatively small misappropriation which is, in reality, a qualitatively vital aspect of the plaintiff's protectable expression. [n.58]

*69 Noting that Judge Pratt had proceeded in the reverse order, the overall analysis at the trial level and the ultimate finding of no copyright infringement due to copying or lack of "substantial similarity" was unequivocally affirmed. [n.59]

Turning to the second issue on appeal, that of the propriety of the preemption according to 17 U.S.C. § 301(a) of the trade secret issue by the copyright issue, Judge Pratt's holding was also affirmed. [n.60] The Court concluded with a summary of its holding as follows:

In adopting the above three step analysis for substantial similarity between the non-literal elements of computer programs we seek to insure two things: (1) that programmers may receive appropriate copyright protection for innovative utilitarian works containing expression; and (2) that non-protectable technical expression remains

in the public domain for others to use freely as building blocks in their own work. At first blush, it may seem counterintuitive that someone who has benefited to some degree from illicitly obtained material can emerge from an infringement suit relatively unscathed. However, so long as the appropriated material consists of non- protectable expression, '[t]his result is neither unfair nor unfortunate. It is the means by which copyright advances the progress of science and art.' Feist, [citation omitted]. [n.61]

Aftermath:

The significance of Computer Associated v. Altai (II) lies in two aspects; first, in dismantling the overextension of copyright protection into the scope of non-literal aspects of computer programs which was the contribution of the decision in Whelan v. Jaslow; second, in providing a detailed test for analyzing two programs wherein the test's point of inquiry requires that the program which was allegedly copied be closely scrutinized to establish what, if any, copyrightable nature subsists in the first program.

In dismantling Whelan v. Jaslow the Court of Appeals for the Second Circuit recognized the many flaws which were inherent in the Whelan v. Jaslow opinion, flaws which existed at the time that the Court of Appeals for the Third Circuit drafted the opinion and which became more pronounced with the passage of time. First, the difference between the "static" and "dynamic" nature of programs was recognized:

The static structure and dynamic structure of a program can be quite different; indeed from dealing with the behavior of a program, i.e., operating it, one can tell virtually nothing about its text. Thus, according to Dr. Davis, it makes no technical sense to talk simply about the "structure" of a program, *70 because the term is ambiguous and the distinction [between dynamic structure and static structure] matters. [n.62]

Such distinction strikes to the heart of the issue of the propriety of copyright protection vis a vis computer programs; namelywhereas the copyright laws have a long and successful history in protecting various forms of artistic and creative expression, the applicability of this form of protection to an intellectual work-product which is dynamic, and clearly more akin to a "... process, system, [and] method of operation ..." is troublesome. In the analysis used by the Court of Appeals for the Third Circuit in Whelan v. Jaslow, the basis of the ultimate inquiry was an inquiry into what was the overall "idea" of the program [n.63] and once the "idea" was defined, an inquiry into the similarity between the "significant steps of the program s " [n.64] was performed. Under this approach, the "idea" was broadly defined as "the efficient management of a dental laboratory" [n.65] and requisite "substantial similarity" was found by a comparison of the written text of the source and object codes *71 of the original program and the infringing program notwithstanding the fact that different programs had been written using different computer languages for different types of computers. [n.66] The finding of "substantial similarity" was based on alleged similarities in the structure of the data file structures [n.67] and in five subroutines. [n.68]

What then, would the result be if the two programs in Whelan v. Jaslow were examined in view of the Second Circuit's three part test of "Abstraction- Filtration-Comparison?"

At the outset, the allegedly copied program would be the initial focus of inquiry, and not the reverse. Properly, the burden of determining which elements of the allegedly copied program should be permitted copyright protection falls upon the alleged victim of copying.

Ignoring the presence of the literal elements of a computer program which are protected under a categorically distinct scope of protection, namely certain types of "screen displays" [n.69] the first step of the test, the "Abstraction" step, requires that the allegedly copied program be separated into its constituent structural parts and then "isolate each level of abstraction in the parts." [n.70] This is essentially the most troublesome part of the three part test as it would be obvious that different individuals [n.71] would separate a program into different parts, and each of the different parts at different "levels of abstraction" along the way. Further, the two endpoints of the test are troublesome as at one extreme, at the "highest" level of abstraction, there would be no copyright protection afforded as that level is pure "idea" (i.e., "the efficient management of a dental laboratory"), and hence unprotectable. At the other endpoint, that at the literal program code level, each computer program language has a unique vocabulary, which is dictated by the constraints of the program language and the computer upon which the program is run. [n.72] Setting the level of abstraction at this "lowest" level would also not provide protection as the copyright would be owned by others (such as in the case of a language compiler) or may be in the public domain, or may be so routine to use in the art that they would *72 be "necessarily incidental" and so "merged" with the underlying idea. [n.73] It would seem most likely that the "level of abstraction" for a program being examined would be at some continuum along the line separating these two endpoints; where along that continuum the "level of abstraction" is established is critical for evaluation of the second and third steps of the test.

Viewing the District Court's record, the expert Dr. Davis has provided a quantified list of the importance of various factors which were exant within a computer program, and that such a list was adopted by the Court, placing a "1000" factor where there were found copied program code sections, but only a "1" relative factor for similar "lists of services" and a "0" factor on the organizational chart of the program. [n.74] Utilizing such a scale in the evaluation of an allegedly copied program tends to indicate that apart from literal copying of the computer code, an overwhelming amount of the non-literal elements of a computer program are strongly related to the underlying purpose of the program, the "idea", or may be the product of the other factors embodied in the second step of the Circuit Court's test for "substantial similarity."

In the second step of the test, "Filtration," the inquiring court would need to examine each structural component at the established "level of abstraction" and determine whether the inclusion at that level was "idea", "dictated by efficiency so to be necessarily incidental to the idea", "required by factors external to the program itself" or from the "public domain." [n.75]

Applying the second step to a "structural component" of the allegedly infringed program with the direction to evaluate its copyrightability will tend to eliminate a great deal of the program being evaluated. Using the highest "level of abstraction" denoted above, the "idea" behind the program would be unprotectable. At the lowest "level", similarly unprotectable as noted above. More likely, an intermediate between the extremes will be established for a "structural component", and for the sake of example we may presume that the structural component "print out a customer's bill" is at such an intermediate level of abstraction. Will it be considered copyrightable as a non-literal element? Looking *73 to the underpinnings [n.76] which the Second Circuit provides for interpretation of its "Filtration" step, chances are minimal.

Can "print out a customer's bill" be considered an idea? Quite possibly yes, as the Circuit Court relies at the genesis of its opinion to the decision of Baker v. Selden which was cited for the proposition that "... an accounting text and a computer program are both 'a set of statements or instructions' ... to bring about a certain result,.... [F]rom this reasoning we conclude that those elements of a computer program that are necessarily incidental to is function are similarly unprotectable." [n.77] By analogy then, the case may be forwarded that this "structural element" of the program is merely an idea, and the execution of the program commands which would gather the necessary information and print out the requisite billing information is necessarily incidental to the practice of the program and the process it provides and thereby unprotectable.

Can the structural element for "print out a customer's bill" be considered "dictated by efficiency so to be necessarily incidental to the idea?" Strong support may be found for such a determination from those elements [n.78] which were elucidated by the Second Circuit. These recited "elements" when reviewed carefully, substantially strip away the effect of the exterior constraints which may be placed on the design of a computer programmer and which limit the degrees of creative freedom which a computer programmer might wish to exert in the design of a program. Once these elements are stripped away, one may then wish to test if the remaining portions of the computer program's structure may have any element which may be subject to copyright protection. Continuing under the guidelines of the "Filtration" step, what remains is to be examined and the applicability of the principle of merger to be reviewed, or those from the public domain. From the Second Circuit's approving commentary regarding the recent holding in Feist v. Rural [n.79] and further in view of the fact that the boundaries of expression upon a computer are limited [n.80], it is very likely that a strong argument may also *74 be made due to the considerations of efficiency and/or the limitations inherent in the programming language, that the various means or necessity to use a "particular set of modules" was dictated by these overriding *75 considerations, and as such would be considered as to have merged with the underlying idea. [n.81] In the alternative, the remaining non-literal aspects of the program may be evaluated to determine if elements and methods of operation are from the public domain or owned by others; such is likely due to the limitations of computer hardware and popular operating systems. [n.82] What may remain, if anything, passes on to the third step, that of "Comparison."

In the last step of the test, "Comparison," the remaining protectable aspects of the original program is examined and compared to the allegedly infringing program. However, the Second Circuit noted in its opinion that the importance of the program's structure was to be deeply discounted as relevance to any determination of "substantial similarity." [n.83]

What does the test provide?

The test provides a rigorous series of filters which are to be used in stripping away any non-literal element which does not fulfill the requirements of 17 U.S.C. § 101 of the alleged original program. The elucidated steps, properly place the initial burden of proof upon the plaintiff's program which is now not presumed to contain any copyright protectable non-literal subject matter until said subject matter is found. Passing through the test, when properly applied, effectively strips away those elements which are notorious in the field of computer programming, those which have passed into the public domain, those which are directed to universally used routines, and those which are considered merged with the underlying idea which is also more narrowly defined at each of the various "levels of abstraction." Any structural element *76 which survives this trial by fire, is then subject to further discounting as is proposed in the calculus of the trial court. Then, and only then, does the inquiry then shift to the allegedly copied program to determine if there had been non-literal copying evidenced by any remaining "substantial similarity."

Surviving the test will not be an easy matter. It is the avowed intention [n.84] of the Second Circuit that surviving the test would be extremely difficult, else such a test would not be proposed, nor would the decision at the trial level be affirmed in its entirety. [n.85] The Second Circuit intended to substantially curtail the applicability of the copyright protection for non-literal elements of computer programs, and that is what they have accomplished by their test, and their clear assignment of the initial burden to the plaintiff's program to prove some essence of copyrightability in the computer program's non-literal structure before a claim of infringement is raised. Further, both the trial court and the appeals court of the Second Circuit have expressly omitted the concept of "look-and-feel" in any analysis, a concept which is increasingly criticized by various courts in their attempt to loosen the Gordian knot binding the copyright law as appropriate intellectual property protection for computer programs. [n.86]

One "bright line" rule which may be understood from the decision is that when a program is expressly rewritten in a good faith effort to remove any computer code which was actually copied, then, that action will be viewed most favorably by the Second Circuit in their assessment of liability for copyright infringement both for past and future acts. This is, however, not to be understood as a gateway for trivial modification of a successful program, as the trial court assessed damages for OSCAR *77 3.4 which was admitted to have copied sections of the computer code of ADAPTER. [n.87] This "bright- line" however provides an equitable remedy in cases such as were detailed in the trial record, and lifts the shadow for computer programmers which work in a closely competitive field.

What next? With the conflicting viewpoints of the Second Circuit and the Third Circuit, and further with the various arguments upon both sides of the issue of copyrightability and computer programs evident both in the decisions of the courts throughout the country and in the scholarly literature, the issue seems ripe for resolution by the Supreme Court. It is time that this thorny issue comes to rest.

[n.a]. Andrew N. Parformak, (FPLC '87) Senior Patent Attorney, Sandoz Inc. The views in this paper are solely those of the author and do not necessarily reflect the views of Sandoz Inc.

[n.1]. Computer Associates Intern., Inc. v. Altai, Inc., 23 U.S.P.Q.2d 1241 (2d Cir.1992)

[n.2]. Computer Associates Intern., Inc. v. Altai, Inc., 775 F.Supp 544, 20 U.S.P.Q.2d 1641 (E.D.N.Y.1991)

[n.3]. 17 U.S.C. § 301(a) reads in pertinent part "... all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106 in works of authorship that are fixed in a tangible medium of expression and come within the subject matter of copyright as specified by sections 102 and 103, whether created before or after that date and whether published or unpublished, are governed exclusively by this title. Thereafter, no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State."

[n.4]. 775 F.Supp at 565, 20 U.S.P.Q.2d at 1655.

[n.5]. Id. at 555, 20 U.S.P.Q.2d at 1647.

[n.6]. Id. at 556, 20 U.S.P.Q. at 1648.

[n.7]. Id. at 556, 20 U.S.P.Q. at 1649.

[n.8]. Id.

[n.9]. Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 230 U.S.P.Q. 481 (3d. Cir.1986), cert. denied, 479 U.S. 1031 (1987).

[n.10]. 797 F.2d 1222, 1235, 230 U.S.P.Q. 481, 491 "In other words, the purpose of 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."

[n.11]. 797 F.2d 1222, 1236, 230 U.S.P.Q. 481, 493-494 "The 'expression of the idea' in a software computer program is the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating and producing information either on a screen, print-out, or by audio communication.... [citation omitted].... We agree. The conclusion is thus inescapable that the detailed structure of the Dentalab program is part of the expression, not the idea, of that program."

[n.12]. Computer Associates Intern., 775 F.Supp. 544, 559, 20 U.S.P.Q.2d 1641, 1650.

[n.13]. Id. at 560, 20 U.S.P.Q.2d at 1651.

[n.14]. Nichols v. Universal Pictures Corp. et al., 45 F.2d 119, 121, 7 U.S.P.Q. 84 (2d. Cir.1930) cert. denied, 282 U.S. 902 (1931).

[n.15]. The Court appointed expert, Dr. Randall Davis of the Massachusetts Institute of Technology, per Federal Rules of Evidence, § 706.

[n.16]. 775 F.Supp. at 562, 20 U.S.P.Q.2d at 1652.

[n.17]. 775 F.Supp. at 563, 20 U.S.P.Q.2d at 1653.

[n.18]. 775 F.Supp. at 565, 20 U.S.P.Q.2d at 1655.

[n.19]. 23 U.S.P.Q.2d 1241 (2d Cir.1992).

[n.20]. Id. at 1245.

[n.21]. 17 U.S.C. § 101 recites "A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."

[n.22]. 23 U.S.P.Q.2d at 1245.

[n.23]. Id. at 1246.

[n.24]. Id. at 1248.

[n.25]. Id. at 1248-1249.

[n.26]. Id. at 1249.

[n.27]. Id. at 1249-1250 where cases cited include Lotus Development Corp. v. Paperback Software International, 15 U.S.P.Q.2d 1577 (D.Mass.1990); Whelan v. Jaslow, supra; as well as others.

[n.28]. 23 U.S.P.Q.2d at 1250.

[n.29]. Id. The Circuit Court further noted: "These items represent products of computer program, rather than the programs themselves and fall under the copyright rubric of audiovisual works. If a computer audiovisual display is copyrighted separately as an audiovisual work, apart from the literary work that generates it (i.e., the program), the display may be protectable regardless of the underlying program's copyright status. ... Of course, the copyright protection that these displays enjoy extends only so far as their expression is protectable."

[n.30]. Id. at 1250. The Court notes the landmark cases of Baker v. Selden, 101 U.S. 99 (1879), and Mazer v. Stein, 347 U.S. 201 (1954) which states the idea-expression separation.

[n.31]. Id. at 1251. Further, the boundary of 17 U.S.C. § § 101(a) and 101(b) is a perilous one, being no less than the boundary of protectable subject matter and unprotectable subject matter.

[n.32]. Id. at 1251.

[n.33]. Supra note 27.

[n.34]. 23 U.S.P.Q.2d at 1251-1252.

[n.35]. Id. at 1252.

[n.36]. Id. at 1252-1253.

[n.37]. Id. at 1253. The Circuit Court further provides guidance in an "anatomical guide" describing its procedure: "At the lowest level of abstraction a computer program may be thought of in its entirety as a set of individual instructions organized into a hierarchy of modules. At a higher level of abstraction, the instructions in the lowest-level modules may be replaced conceptually by the functions of those modules. At progressively higher levels of abstraction, the functions of high-level modules conceptually replace the implementations of those modules in terms of lower-level modules and instructions, until finally, one is left with nothing but the ultimate function of the program."

[n.38]. 23 U.S.P.Q.2d at 1253.

[n.39]. Id.

[n.40]. Id. at 1254. The Circuit Court cites Morrissey v. Procter & Gamble Co., 379 F.2d 675, (1st Cir.1967), Concrete Machinery Co. v. Classic Lawn Ornaments, Inc. 843 F.2d 600 (1st Cir.1988) and Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738 (9th Cir.1971) to note that under the doctrine of merger copyright will not be extended where there are one or very limited methods of expressing an idea, then these expressions are considered "merged" with the idea itself.

[n.41]. Id. at 1254 (emphasis in original).

[n.42]. Id. at 1254-1255.

[n.43]. Id. at 1255.

[n.44]. Plains Cotton Cooperative Association of Lubbock, Texas v. Goodpasture Computer Service, Inc., 807 F.2d 1256, 1262, 1 U.S.P.Q.2d 1635 cert. denied. 484 U.S. 821 (1987).

[n.45]. Manufacturer's Technologies, Inc. v. Cams, Inc., 706 F.Supp. 984, 995-999, 10 U.S.P.Q.2d 1321, 1329-1334 (D.Conn.1989), Data East U.S.A, Inc. v. Epyx, Inc., 862 F.2d 204, 209, 9 U.S.P.Q.2d 1322 (9th Cir.1988).

[n.46]. Q-Co Industries, Inc. v. Hoffman, 625 F.Supp. 608, 616, 228 U.S.P.Q. 554, 559 (S.D.N.Y.1985).

[n.47]. Computer Associates Intern., 23 U.S.P.Q.2d at 1256.

[n.48]. Id.

[n.49]. Id. The Circuit Court also noted from Data East U.S.A., Inc. v. Epyx, Inc., 862 F.2d at 208, dicta further focusing on the analysis to be performed at this third step: "To determine whether similarities result from unprotectable expression, analytic dissection of similarities may be performed. If ... all similarities in expression arise from use of common ideas, then no substantial similarity can be found."

[n.50]. 23 U.S.P.Q.2d at 1256, "... CA and some amici argue against the type of approach that we have set forth on the grounds that it will be a disincentive for future computer program research and development. At bottom, they claim that if programmers are not guaranteed broad copyright protection for their work, they will not invest the extensive time, energy and funds required to design and improve program structures. While they have a point, their argument cannot carry the day. The interest of the copyright law is not in simply conferring a monopoly on industrious persons, but in advancing the public welfare through rewarding artistic creativity, in a manner that permits the free use and development of non-protectable ideas and processes."

[n.51]. Feist Publications, Inc. v. Rural Telephone Service Co., Inc., 499 U.S. 340, 18 U.S.P.Q.2d 1275 (1991).

[n.52]. 499 U.S. at 1297, 18 U.S.P.Q.2d at 1285.

[n.53]. 23 U.S.P.Q.2d at 1257. By this analogy, the Circuit Court opens a whole new vista upon the area of copyright and protection utilizing the underpinnings of the Supreme Court's decision in Feist Publications, Inc. v. Rural Telephone Service Co., Inc. In the case, Rural alleged copyright infringement when Feist copied telephone subscriber information for use in its telephone directory which covered a larger geographical area than Rural's, and independently added information thereto. The issues to be decided were the boundaries of two well-established propositions, namely (i) facts were not copyrightable, and (ii) compilations of facts were generally copyrightable.

Writing for the majority, Justice O'Connor established the boundaries noting: "Copyright treats facts and factual compilations in a wholly consistent manner. Facts, whether alone or as part of a compilation, are not original and therefore may not be copyrighted. A factual compilation is eligible for copyright if it features an original selection or arrangement of facts, but the copyright is limited to the particular selection or arrangement. In no event may copyright extend to the facts themselves." 499 U.S. at 350, 18 U.S.P.Q. at 1280.

O'Connor further went on to note that Congress in the 1976 revision of the copyright act had added § 102(b) which excluded:

"... idea, procedure, process, system, method of operation, concept, principle or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work" noting that § 102(b) was to be universally understood as to "... prohibit any copyright in facts" and to further recognize that the definition of "compilation" in 17 U.S.C. § 101 to read "a work formed by the collection and assembly of preexisting materials or of data that are selected, coordinated, are arranged in such a way that the resulting work as a whole constitutes an original work of authorship" (emphasis in original). 499 U.S. at 356, 18 U.S.P.Q.2d at 1282.

Justice O'Connor continued on to state:

"The statute identifies three distinct elements and requires each to be met for a work to qualify as a copyrightable compilation: (1) the collection and assembly of pre-existing material, facts, or data; (2) the selection, coordination, or arrangement of those materials; and (3) the creation, by virtue of the particular selection, coordination, or arrangement, of an 'original' work of authorship." 499 U.S. at 357, 18 U.S.P.Q.2d at 1282.

Clarifying this statement,

"The key to the statutory definition is the second requirement. It instructs courts that, in determining whether a fact-based work is an original work of authorship, they should focus on the manner in which the collected facts have been selected, coordinated, and arranged. This is a straight-forward application of the originality requirement. Facts are never original, so the compilation author can claim originality, if at all, only in the way the facts are presented. To that end, the statute dictates that the principal focus should be on whether the selection, coordination, and arrangement are sufficiently original to merit protection.

Not every selection, coordination, or arrangement will pass muster. This is plain from the statute. It states that, to merit protection, the facts must be selected, coordinated, or arranged 'in such a way' as to render the work as a whole original. This implies that

some "ways" will trigger copyright, but others will not. [W]e conclude that the statute envisions that there will be some fact-based works in which the selection, coordination, and arrangement are not sufficiently original to trigger copyright protection." 499 U.S. at 358, 18 U.S.P.Q.2d at 1283.

[n.54]. 23 U.S.P.Q.2d at 1257.

[n.55]. 23 U.S.P.Q.2d at 1257. The Circuit Court noted that patent protection might be more applicable to the area of intellectual property protection of computer programs.

[n.56]. 23 U.S.P.Q.2d at 1257. "While incentive based arguments in favor of broad copyright protection are perhaps attractive from a pure policy perspective, [citation omitted] ultimately, they have a corrosive effect on certain fundamental tenents of copyright doctrine. If the test we have outlined results in narrowing the scope of protection, as we expect it will, that result flows from applying, in accordance with Congressional intent, long- standing principles of copyright law to computer programs. Of course our decision is also informed by our concern that these fundamental principles remain undistorted."

[n.57]. 23 U.S.P.Q.2d at 1258.

[n.58]. 23 U.S.P.Q.2d at 1259.

[n.59]. 23 U.S.P.Q.2d at 1259-1260.

[n.60]. 23 U.S.P.Q.2d at 1260-1262.

[n.61]. 23 U.S.P.Q.2d at 1262.

[n.62]. 20 U.S.P.Q.2d at 1650. Judge Pratt further noted at pp. 1650-1651:

"Going beyond Dr. Davis' analysis, the court notes a possible statutory difficulty that arises when we recognize, as we must, that a computer program 'behaves'. Section 102(a) provides that copyright protection subsists for original works of authorship including those making up computer programs. Subsection (b) provides, however: 'In no case does copyright protection for an original work or authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such

work.' (emphasis added). Since the behavior aspect of a computer program falls within the statutory terms 'process', 'system', and 'method of operation', it may be excluded by statute from copyright protection. Indeed it has been suggested that computer software is better protected by patent law than by copyright law.... Fortunately, this court need not wrestle with that possible development in the law of intellectual property, because CA's rights in this case are fully protected by viewing the ADAPTER program at text."

[n.63]. 230 U.S.P.Q. at 492-493. "The 'expression of the idea' in a software computer program is the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating, and producing information either on a screen, print-out or by audio communication. [citation omitted] We agree. The conclusion is thus inescapable that the detailed structure of the Dentalab program is part of the expression, not the idea, of that program." (emphasis added).

[n.64]. 230 U.S.P.Q. at 498. "Computer programs are no different. Because all steps of a computer program are not of equal importance, the relevant inquiry cannot therefore be the purely mechanical one of whether most of the programs' steps are similar. Rather, because we are concerned with the overall similarities between the programs, we must ask whether the most significant steps of the programs are similar."

[n.65]. Supra note 33.

[n.66]. 230 U.S.P.Q. at 483.

[n.67]. 230 U.S.P.Q. at 495.

[n.68]. 230 U.S.P.Q. at 498.

[n.69]. 23 U.S.P.Q.2d at 1250.

[n.70]. Supra note 36.

[n.71]. Read: "judges, experts, plaintiff's counsel, defendant's counsel."

[n.72]. Of course, it is to be understood that different versions of popular programming languages may share the same or subsets of common vocabulary, such as the case with the popular programming languages "C" and its progeny "C+" and C++."

[n.73]. For example, whereas there are myriad of computer languages which specify a different command to print a substring in a line of text, each of these different commands performs the same task; in each computer language there is only one (or few) commands to perform the task, and due to sake of efficiency, only one or two are necessary. The command merges with the idea.

[n.74]. 20 U.S.P.Q.2d at 1652.

[n.75]. 23 U.S.P.Q.2d at 1253.

[n.76]. Id. at 1253.

[n.77]. 23 U.S.P.Q.2d at 1251.

[n.78]. 23 U.S.P.Q.2d at 1255.

[n.79]. See n. 53.

[n.80]. See Lotus Development Corp. v. Paperback Software International, 15 U.S.P.Q.2d 1577 (D.Mass.1990) In the case, literal elements of Lotus' 1-2-3 program were alleged to have been copied by Paperback's VP-Planner. There, Judge Keeton noted: "I credit the testimony of expert witnesses that the bulk of the creative work is in the conceptualization of a computer program and its user interface, rather than in its encoding, and that creating a suitable user interface is a more difficult intellectual task, requiring greater creativity, originality, and insight, than converting the user interface design into instructions to the machine." Id. at 1589-1590. Later in the opinion, when viewing what common features were shared in several electronic spreadsheet programs also stated:
"As already noted, these three products--VisiCalc, 1-2-3, and Excel--share the general idea of an electronic spreadsheet but have expressed the idea in substantially different ways. These products also share some elements however, at a somewhat more detailed or specific point along the abstractions scale. One element shared by these and many other programs is the basic spreadsheet screen display that resembles a rotated "L".... Although Excel used a different basic spreadsheet screen display that more closely

resembles a paper spreadsheet, there is a rather low limit, as a factual matte, on the number of ways of making a computer screen resemble a spreadsheet. Accordingly, this aspect of electronic spreadsheet computer programs, if not present in every expression of such a program, is present in most expressions. Thus, the second element of the legal test weighs heavily against treating the rotated "L" screen display as a copyrightable element of a computer program....

Another expressive element that merges with the idea of an electronic spreadsheet--that is an essential detail present in most if not all expressions of an electronic spreadsheet--is the designation of a particular key that, when pressed, will invoke the menu command system. The number of keys available for this designation is limited for two reasons. First, because most of the keys on the keyboard relate either to values (e.g., the number keys and the mathematical operation keys) or labels (e.g., the letter keys), only a few keys are left that can be used, as a practical matter, to invoke the menu command system. Without something more, the programmed computer would interpret the activation of one of these keys as an attempt by the user to enter a value or label into a cell.... As just noted, when all the letter, number, and arithmetic keys are eliminated from consideration, the number of keys remaining that could be used to invoke the menu command system is quite limited. They include the slash key ("/") and the semicolon key (";"). The choice of the creators of VisiCalc to designate the slash ("/") key to invoke the menu command system is not surprising.... Each of the elements just described is present in, if not all, at least most expressions of an electronic spreadsheet computer program. Other aspects of these programs, however, need not be present in every expression of an electronic spreadsheet. An example of distinctive details of expression is the precise 'structure, sequence, and organization,' Whelan, 797 F.2d at 1248, of the menu command system.

The fact that these specific command terms are quite obvious or merge with the idea of such a particular command term does not preclude copyrightability for the command structure taken as a whole.... A different total structure may be developed even from individual components that are quite similar and limited in number. To determine copyrightability, a court need not--and, indeed, should not--dissect every element of the allegedly protected work. Rather the court need only identify those elements that are copyrightable, and then determine whether those elements, considered as a whole, have been impermissibly copied: 15 U.S.P.Q.2d at 1598-1599.

[n.81]. 23 U.S.P.Q.2d at 1254. Unless of course, the programmer elects to write computer code which is non-optimal in contrast to the "industry wide goals" and accept the repercussions such as slower response, larger program size and memory requirements.

[n.82]. For example, on the IBMR and MacintoshR computers, most of the operations are performed by clicking upon a visual marker on the screen to initiate actions, or by clicking-and-dragging part of a screen display to another part of the screen, so to effect a response from the program. Further visual elements include a horizontal or vertical bar which has "buttons" which when clicked, elicit a response. These visual elements are used, for example, in popular word processing programs for the Windows* operating

system for the IBMR personal computer. Thus, for all practical purposes, the industry has accepted the use of such visual tokens as the most desirable, and "efficient" form of program design; in the alternative, there are relatively limited alternatives to design such visual tokens other than their shape, size and arrangement, there are functional equivalents of these tokens for common operations which exist in each of these programs. Further, as industry develops, more standardization of the "user interface" is to be expected as it has long been an industry goal to provide "easy-to-use" programs.

[n.83]. Supra note 16. It is quite foreseeable in many instances, that nothing will survive.

[n.84]. 23 U.S.P.Q.2d at 1249-1250. The court notes:

"The syllogism that follows from the foregoing premises a powerful one; if the non-literal structures of literary works are protected by copyright; and if computer programs are literary works, as we are told by the legislature; then the non-literal structures of computer programs are protected by copyright. [citations omitted] We have no reservation in joining the company of those courts that have already ascribed to this logic. [citations omitted] However, that conclusion does not end our analysis. We must determine the scope of copyright protection that extends to a computer program's non-literal structure."

[n.85]. 23 U.S.P.Q.2d at 1245.

[n.86]. 15 U.S.P.Q.2d at 1595. "Despite its widespread use in public discourse on the copyrightability of nonliteral elements of computer programs, I have not found the 'look and feel' concept, standing alone, to be significantly helpful in distinguishing between nonliteral elements of a computer program that are copyrightable and those that are not."

[n.87]. 20 U.S.P.Q. at 1660.