Softel, Inc. v. Dragon Medical and Scientific Communications, Inc.

118 F.3d 955 (2d Cir. 1997), **cert. denied,** *523 U.S. 1020 (1998)*

[Softel and Dragon had a longstanding relationship under which Softel provided video imaging software to Dragon. Softel owned the copyright in the code it supplied. The relationship soured. Dragon retrieved files that Softel had attempted to delete from Dragon's system and used Softel's image retrieval modules to create new programs. After Softel sued for

copyright infringement, Dragon created new programs written in a different language to run on different hardware (the "post-litigation" programs). The judge below found that the pre-litigation programs infringed Softel's copyrights, but the post-litigation ones did not. The appeal addresses Softel's claims regarding the post-litigation programs.]

PARKER, J.: . . . On this appeal, Softel renews its claim that Dragon's post-litigation programs infringed copyrights held by Softel. Softel argues that it claimed in the district court that its software combined certain computer programming design elements in an expressive way and that Dragon had copied that expression, but that Judge Cannella ignored this claim and instead addressed (and rejected) a claim that Softel had not made, to wit, that each of the design elements, *taken individually*, was protectible expression.

It is well-established in this Circuit that non-literal similarity of computer programs can constitute copyright infringement. *See Altai*, 982 F.2d at 702. We have prescribed an "abstraction-filtration-comparison" method of analysis for determining whether a program has been infringed. *See id.* at 706-12. . . .

Our application of this method of analysis to the facts of *Altai* demonstrates that an allegation of infringement based on similarities in architecture cannot be ignored merely because many or all of the design elements that make up that architecture are not protectible when considered at a lower level of abstraction. In *Altai*, the district court held many aspects of the program at issue in that case to be not protectible for various reasons (e.g., because they were in the public domain or were computer *scenes a faire*). Nevertheless, the court proceeded to a higher level of abstraction and responded to the plaintiff's claim of infringement based on alleged similarities between the two programs' "organizational charts." *Computer Assocs. Int'l. Inc. v. Altai, Inc.*, 775 F. Supp. 544, 562 (E.D.N.Y. 1991). In reviewing that claim, the trial court made no effort to remove from its analysis those elements of the program that had been found unprotectible at the lower level of abstraction. Instead, analyzing the program at the higher level of abstraction, it rejected the organizational claim on the grounds that the structure alleged to have been infringed was "simple and obvious to anyone exposed to the operation of the program[s]." *Id.* This Court approved that approach, adding only that the district court's use of the word "obvious" should be understood to be a holding that the purportedly infringed structure was a *scene a faire*. *See Altai*, 982 F.2d at 714-15.

The foregoing approach is consistent with the Supreme Court's decision in *Feist Publications, Inc. v. Rural Telephone Service Co.*, 499 U.S. 340 (1991), discussed by this Court in *Altai*, 982 F.2d at 711-12. In *Feist*, the Court made quite clear that a compilation of non-protectible elements can enjoy copyright protection even though its constituent elements do not. *See Feist*, 499 U.S. at 344-51 A district court in another circuit has illustrated the danger of overlooking this aspect of copyright law with an astute hypothetical:

> Suppose defendant copied plaintiff's abstract painting composed entirely of geometric forms arranged in an original pattern. The alleged infringer could argue that each expressive element (i.e., the geometric forms) is unprotectible under the functionality, merger, scenes a faire, and unoriginality theories and, thus, all elements should be excluded prior to the substantial similarity of expression analysis. Then, there would be nothing left for purposes of determining substantial similarity of expression. In this example, elimination of "unprotectible" elements would result in a finding of no copyright infringement, which would be clearly inconsistent with the copyright law's purpose of providing incentives to authors of original works.

Apple Computer, Inc. v. Microsoft Corp., 779 F. Supp. 133, 136 (N.D. Cal. 1991), aff'd in relevant part and rev'd and remanded in part, 35 F.3d 1435, 1444 (9th Cir. 1994). Scholars agree: ["] . . . An original arrangement of uncopyrightable or public domain works — even

facts — is as copyrightable as a compilation in the computer context as it is elsewhere in copyright law. Thus, individual program elements that are "filtered" out at one level may be copyrightable when viewed as part of an aggregate of elements at another level of abstraction.["] Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977, 1003 (1993) (footnote omitted); *see also* 3 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* §13.03[F][5], at 13-145 (1996) ("*Nimmer*") ("In performing the filtering, the court should be sensitive to the myriad ways in which copyrightable creativity can manifest itself; the analysis should not proceed mechanically simply by isolating physical elements out of the copyrightable work."). Indeed, when applied to the context of literary works, this point becomes quite obvious: taken individually, the words that constitute a literary work are not copyrightable, yet this fact does not prevent a literary text, i.e., a collection of words, from enjoying copyright protection. *See Nimmer* §13.03[F][5], at 13145 n. 345.1 (explaining that the fact that Hamlet's soliloquy can be atomized into unprotectible words does not mean that the soliloquy as a whole lacks originality for copyright purposes). . . .

[Softel's president] testified regarding relationships between all four of the design elements at issue: apparently, the menus cued external files which read English language commands into the main program, where the commands cued modules of code. Standing alone, these allegations may not help answer the question of how many ways existed to design a computer program with the same functionality as Softel's. However, when they are combined with an allegation that the specific commands used by Dragon were nearly identical with Softel's, they do appear to establish at least a colorable claim that there was a modicum of expression in the design of the program and that Dragon infringed that expression. That is, even if there were few ways to design such a program, it does not seem likely that Dragon would have to use an identical structure *and* copy approximately fifteen out of fifteen commands.

Softel presented an argument, supported by some evidence, that the manner in which it had combined certain computer design elements was expressive for purposes of copyright law, and that Dragon had copied this expression. Keeping in mind that the requisite level of originality for copyright protection is "minimal" and "extremely low," *Feist*, 499 U.S. at 345 we cannot say that Softel's claim was without merit on its face. However, several aspects of Judge Cannella's opinion reveal that he either ignored or misanalyzed Softel's argument, and consequently failed to perform the *Altai* analysis at the highest level of abstraction — here, the interrelationships among the four identified elements. . . .

[The district] court's findings of fact and conclusions of law address each element individually, but not in relation to each other. For example, the court credited Dragon's expert testimony that each of the four design elements was pervasively used in the computer industry, but did not address the issue of whether the choice and manner of combination of the four elements was commonplace. Similarly, the court found that the programmer at Dragon who was responsible for the post-litigation programs had learned to use external files from another source, and that certain code is required whenever a screen touchpoint is used, but did not address the question of where Dragon's programmer learned to combine external files, or touch-point finger-finding algorithms, with the other design elements. . . .

The district court's dismissal of Softel's claim [ ] of non-literal infringement . . . [is] vacated and remanded for further consideration consistent with this opinion. . . .

NOTES AND QUESTIONS

1. Does the court's decision in *Altai* make theoretical sense? Does it make technological sense? Do you think courts will be able to implement it as a practical matter? Is it an

improvement over the *Whelan* test to which it refers? Would you argue that it, like *Hoehling* (review note 3, *supra* page 93), is about the cost of errors?

2. The *Altai* court points out that much, if not most, of the investment in creating a program is made in designing it. Actual coding is not that pricey. If *Altai* practically means that copyright protects the literal code but little of the design structure, is this result consistent with the policy bases for copyright discussed in Chapter 1? How does the *Altai* court respond to the argument that its decision provides "too little" protection to program design?

3. Does the syllogism employed by the *Altai* and *Whelan* courts — computer programs are protected as literary works; literary works are protected against nonliteral infringement; therefore, computer programs are protected against nonliteral infringement — make sense? Consider the following:

> The copyright question is usually posed as whether [Sequence, Structure, and Organization (SSO)] is protected as a "nonliteral element" of the protected program code. The analogy is to the copyright in a novel or play, which has long been recognized to extend beyond the verbatim language to more or less detailed elements of plot sequence. Most of the proponents of broad program copyrights conveniently forget that copyright in other types of literary works, such as histories, fact works, rule books, and technical works, is much "thinner," limited to verbatim language and close paraphrases thereof. They also usually forget that computer programs are only literary works in form; in substance they are the technology for using computers. Consequently, reasoning by analogy to traditional (nonfunctional) works without resort to policy cannot be expected to lead to sensible results, unless one is a strong believer in luck.

Dennis S. Karjala, *The Relative Roles of Patent and Copyright in the Protection of Computer Programs*, 17 J. Marshall J. Computer & Info. L. 41, 53 (1998).

Professor Karjala argues that policy considerations do not support copyright protection for a program's structure. Nevertheless, he applauds the *Altai* decision. In his view, "[i]n fact, a court that applies the *Computer Associates* filters honestly will soon realize that *everything* in the SSO is present for the purpose of making the program function better, that is, for efficiency reasons. Consequently, under *Computer Associates*, after filtering for efficiency there is very little, if anything, to protect besides the code." *Id.* at 54.

Do you agree? Are efficiency considerations monolithic? Put differently, programmers make efficiency trade-offs. Some might choose to optimize use of memory; others the usability of the program. Program structure varies depending on those choices. Recall the Clapes excerpt, *supra* page 217. Do such choices evidence creativity and justify copyright protection?

4. Is the court's decision in *Softel* consistent with Professor Karjala's interpretation of *Altai* and its likely consequences for copyright protection of program structure? Is *Softel* consistent with *Altai* generally? Does it make sense to treat the structure of a program as a compilation? (Review note 9 on page 116 discussing the *Atari Games Corp. v. Oman* case.)

5. Recall the *Franklin* case, *supra*, and the last sentence of the case excerpt: "Franklin may wish to achieve total compatibility with independently developed application programs written for the Apple II, but that is a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged." *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984). The court in *Altai* indicated that elements of a program required for compatibility should be filtered out. Is this consistent with *Franklin*?

Alternatively, might compatibility considerations preclude copyrightability altogether rather than serving as one "filter" in the infringement analysis? In *Lexmark International, Inc. v. Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004), the court addressed

the copyrightability of a short program consisting of only eight commands. Noting the briefness of the plaintiff's program, the court observed that "unless a creative flair is shown, a very brief program is less likely to be copyrightable because it affords fewer opportunities for original expression." *Id.* at 542-43. The court found that plaintiff's code functioned as a "lock-out" code that prevented communication between two devices unless the exact code was copied. The court held that "[t]o the extent compatibility requires that a particular code sequence be included in the component device to permit its use, the merger and scenes a faire doctrines generally preclude the code sequence from obtaining copyright protection." *Id.* at 536. Which approach is better — *Franklin*'s, *Altai*'s, or *Lexmark*'s?

6. From whose perspective should compatibility requirements be assessed? In *Dun & Bradstreet Software Services, Inc. v. Grace Consulting, Inc.*, 307 F.3d 197 (3d Cir. 2002), *cert. denied*, 538 U.S. 1032 (2003), the defendant, a computer consulting corporation formed to provide services to plaintiff's licensees, began to offer competing software programs to the licensees. Defendant used copy and call commands to access the plaintiff's software so that its programs would interoperate with the plaintiff's program. It argued that industry practice justified duplication of copy and call commands for this purpose. The court rejected the argument, holding that in determining whether program elements are dictated by external factors, including interoperability, a court must examine the program from the viewpoint of its creator, not that of the alleged infringer. Is this consistent with *Altai*? What policy goals might be furthered by using the alleged infringer's viewpoint rather than the creator's?

7. Note that a court's decision about what elements of a work are copyrightable influences the answer to the ultimate question whether infringement has occurred. We revisit this issue in Chapter 5.