Overview of Recent Developments in Copyright Protection for Software

Bradley J. Hulbert
Allegretti & Witcoff, Ltd.
Chicago and Boston

1 Introduction

Section 106 of Title 17 of the United States Code prohibits the unauthorized preparation of "derivative works based upon ... copyrighted work." Copyrights, unlike patents, protect against copying. [1] Two authors may properly create and use virtually the same work, if they create their works independently. However, when one work was prepared before the other, the author of the second work had access to the prior work, and the works are substantially similar, the courts may infer the later work is an impermissible "derivation" of the prior work.

Much recent litigation regarding copyrights for software centers on whether or not a later-developed program, which has no statements or elements which are literally the same as those in a prior program, may nonetheless be a "derivative work." When required to decide whether a later-developed program was impermissibly "derived" from a previous work, courts often face a difficult task. Most judges and jurors have, at best, only a rudimentary understanding of programming languages. Accordingly, they must depend upon the testimony of paid experts regarding whether or not the structure, sequence, and operation (or "look and feel") of two programs are "substantial similar." The courts have adopted a variety of analytical techniques for determining whether a later program is a non-literal, but impermissible, copy. Four major techniques used by the courts are discussed below.

[1] Patents, in contrast, protect against the unauthorized making, using, or selling of a patented invention. Thus, if an inventor independently develops an invention that is clearly within the boundaries of a claim of a valid and enforceable patent, the making, using, or selling of the device by the second inventor is prohibited. This prohibition exists regardless of whether or not the second inventor made the later device independently or by copying. If the second inventor did make the infringing device through copying, however, courts may, for example, award treble damages.

The U.S. Supreme Court has decreed that most computer programs are protectable by patents in the landmark decision Diamond v. Diehr, 450 U.S. 175 (1981). See generally, Hulbert, Special Considerations for Obtaining and Litigating Software Patents, 4 Software Law Journal 1 (1990). The liability of the independent software developer for infringement has caused increasing concern. See, e.g., Software Flood the Patent Office, Infoworld at p. 43 (Sept 30, 1991).

2 Analytical Approaches of the Courts

2.1 Distinguishing between an idea and an expression

Section 102(b) of Title 17 provides that only expressions of ideas, rather than ideas themselves, are proper subject matter for copyright protection. Thus, in deciding whether a second work is an impermissible copy, courts frequently look to whether the "ideas" or "facts" have been misappropriated or whether the "expressions" of the ideas or facts have been appropriated.

For example, earlier this year, in Fiest Publications v. Rural Telephone Service Co., Inc., 111 S Ct 1282 (1991), the U.S. Supreme Court decided a landmark "fact vs. expression" case. the defendant, Rural telephone, had produced a white pages telephone directory, listing the name, address, and telephone number of the residents in a community. The plaintiff, Fiest, was alleged to have infringed the copyright on the telephone directory by copying some of the names, addresses, and phone numbers listed in the defendant's directory in order to publish its own, competing directory.

The U.S. Supreme Court reinforced that ideas and facts themselves are never copyrightable. The Court confirmed that a work must be an original work of authorship to qualify for copyright protection. No one may claim originality as to facts, since facts do not owe their origin to an act of authorship.

Court held in Fiest that the actual names, addresses, and telephone numbers of the persons in the community are uncopyrightable facts, and found Fiest not liable for copyright infringement. The Court made clear, however that the particular selection and arrangement of facts in a directory may indeed be copyrightable subject matter.

When applied to copyright protection for software, the Supreme Court's reasoning in Fiest indicates that the raw information in data bases may not be protectable. However, the format in which those basic data are arranged and the selection of particular data which are useful to a user may indeed be copyrightable.

Similarly, the idea of using a spread sheet itself is not copyrightable. Nonetheless, the particular format and sequences used may be protected by copyright.

While the "idea versus expression" formula may assist in the most obvious cases, the distinction between ideas and expressions is often murky. For example, a computer program subroutine to speed the operation of the computations may be considered both an "expression" of how a programmer wants a particular program to perform and an "idea" to enhance a computer's capability. Accordingly, other courts have expressed their reasoning differently.

2.2 The Wheland test: What is necessary to the "purpose or function" of the program.

In Wheland Associates v. Jaslow Dental Laboratories Inc., 797 F.2d 1223 (3rd Cir 1986), the Court of Appeals for the Third Circuit expressed the ideas versus expression dichotomy somewhat differently, stating:

> The purpose or function of a utilitarian work would be in the work's idea and everything that is not necessary to that purpose or function would be part of the expression of the idea... Where there are various means of achieving the desired purpose, then the

> particular means chosen is not necessary to the purpose; hence there is expression, not idea.

Id. at 1236. Similarly, in Lotus Development Corp v. Paperback Software International, 740 F. Supp 37 (D. Mass 1990), the U.S. District Court for the District of Massachusetts proclaimed: "The decision maker must focus on whether those elements œallegedly copied— are a substantial part of the allegedly copyrightable work." Id. at 61.

Thus, under the Wheland test, the court would conclude that it was necessary for a spread sheet program to use columns, and, accordingly, this is an unprotectable aspect of the program. Always printing, for example, a computer's computation time at the bottom left of any print out would not be necessary to the function or purpose of a spread sheet program and, thus, may embody protectable expression under the Wheland test. Questions remain unresolved, however, since most program statements can be considered, at some level, "necessary" to the program; otherwise the statements would not be in the program.

2.3 Limitations on alternatives: Scenes a faire.

Other courts have recognized that certain elements of a program are essential to the basic purpose of a function of a program. Thus, for example, in writing a "western," it is expected that the primary characters will ride horses and carry six-bullet side arms. Similarly, in a program, to the extent that certain aspects of the program are required for the program to operate in a manner generally expected by users, such aspects are not copyrightable expression.

Thus, for example, a spread sheet program is expected to have columns and allow the user to easily add up the numbers in the rows and columns and to provide an easy way of putting in a formula by which the numbers in the spread sheet may be manipulated. These are essential parts, or scenes of faire, of any spread sheet program and thus is not copyrightable. Creative approaches to implement such general ideas, however, may well be copyrightable. Again, however, much disagreement exists, even among experts, regarding what is an essential part of a program.

2.4 Level of abstraction test

Other courts use the so-called level of "abstraction test." As set forth in Health Care Affiliated Services v. Lippiney, 701 F. Supp 1142 (W. D. Penn 1988):

> The evidence merely documents that certain choices were made among factors at a gross level, e.g., the scope of the system, the number of variables to be used or the portions of the work force to be included in the calculation of labor hours. The result of these choices, however, does not constitute program structure sequence and operation.

Under this approach, the courts attempt to state the overall purpose of a program and, to the extent that purpose is abstract (e.g., "spread sheet program") the more likely it is that the program is unprotectable under copyright law. To the extent that the purpose of a particular subroutine is specific (e.g., "increase computation speed by transferring the contents of one particular file to another only if a predetermined condition is met"), the routine is more likely to be protectable under copyright law.

3 Conclusion

Overall, the courts' various approaches to determining whether a particular program is an improper "derivative" work all relate to the same basic premise: to the extent that an author has used creativity to express a function not apparently essential to the overall purpose of a program, the organization (non-literal elements) of the program is protectable. To the extent that the choices made by the programmer were dictated by external concerns (such as designating the largest key on the key board as the one to press in order to enter information), such choices should not, as a matter of public policy, be protected by the copyright law.

In all events, however, the courts encourage independent authorship. Even the more mundane and obvious procedures in a program may be considered improper copies if a programmer, rather than writing his or her own code, clearly takes the "short cut" of simply duplicating the source code of another.