January 1993

Software Reverse Engineering and Cleanrooming, When Is It Infringement?

Jonathan Owens

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/vol9/iss2/6

This Comment is brought to you for free and open access by the Journals at Santa Clara Law Digital Commons. It has been accepted for inclusion in Santa Clara High Technology Law Journal by an authorized administrator of Santa Clara Law Digital Commons. For more information, please contact sculawlibrarian@gmail.com.
SOFTWARE REVERSE ENGINEERING AND CLEAN-ROOMING, WHEN IS IT INFRINGEMENT?

Jonathan Owens†

TABLE OF CONTENTS

I. INTRODUCTION ...................................... 528
II. BACKGROUND ....................................... 530
   A. CONTU and the Addition of § 117 ............... 531
   B. Extension of Copyright Protection to Computer Programs .............. 532
   C. Copyright Infringement .......................... 534
   D. Protection of Computer Programs by Patent ............. 535
III. THE CLEAN ROOM .................................. 537
IV. CASE DECISIONS INVOLVING REVERSE ENGINEERING .................. 539
   A. E.F. Johnson Co. v. Uniden Corp. of America .... 539
   B. Hubco Data Products Corp. v. Management Assistance Incorporated ............................................. 540
   C. Sega Enterprises, Ltd. v. Accolade, Inc. .......... 540
V. APPLICATION OF THE LAW .............................. 543
   A. What can a company which develops and manufactures software do to protect its programs from being copied? ................. 543
   B. What can a company which practices disassembly do to protect itself? .................. 544
VI. SOLUTION AND PROPOSAL .......................... 545
VII. CONCLUSION ........................................ 549

Copyright © 1993 by Jonathan Owens.
† B.S.E.E. Montana State University; J.D. Santa Clara University School of Law 1993. Mr. Owens is currently a technical consultant with the San Francisco intellectual property law firm of Haverstock, Medlen & Carroll.
I. INTRODUCTION

The computer software industry, like many other high-tech industries, is extremely competitive. Development costs of computer programs are far greater than the cost of their duplication. As computers become more commonplace and their potential uses more numerous, the level of competition can only increase. To compete in this industry many software companies have resorted to analyzing their competitor's programs by reverse engineering, which leads to copying of portions of code and in some instances, whole programs. Reverse engineering allows the second company to cut its development costs at the expense of the first company. Another method used by companies to capture a share in this competitive market is to copy only the most valuable aspects of a program, such as the interface between the program and the user.

Understandably, software manufacturers who are developing programs wish to protect their investment and ingenuity. Copying of computer programs not only deprives the author of the chance to recover expenses and gain a reward for his hard work and creativity but also serves to detract from the value of such programs. While copyright law will protect some aspects of computer programs, independent creation is still an affirmative defense to an infringement claim. But there seems to be disagreement about just how far a company can go in the reverse engineering process and how much of the original program can legally be utilized in a subsequent program.

From the beginning, programmers have incorporated other peoples' ideas and adapted them for specific projects. It is this interaction and free exchange of ideas that has allowed the American software industry to become the best in the world. But, because

1. NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT AND RECOMMENDATIONS, 26 (1978).
3. In Lotus Dev. Corp. v. Paperback Software Int'l, 740 F.Supp. 37, 70 (D. Mass. 1990), Paperback developed a spreadsheet program very similar to Lotus' 1-2-3, and even went so far as to copy the user interface and commands from Lotus' program.
the industry has become very competitive, software companies are realizing that to survive, it is necessary to protect their investment and enforce their copyrights.8

Reverse engineering is a process in which a finished program is dissected, decompiled, or downloaded.9 The code is then analyzed and flow charted to discover the method and technique that was utilized in the creation of that program. When a computer program is reverse engineered, it is taken from its finished state and analyzed until it is determined how the program was put together, and why it was done that way.10 A company's reasons for reverse engineering include investigating new innovations used by competitors and the desire to achieve compatibility with a competitor's product. The clean-room is a technique used by a company to attempt to insulate itself from the legal liability that may follow from using portions of another's program.11 A company employing this technique will have a first group of employees reverse engineer a program, in order to obtain the specifications and method of that program. These specifications and technique will then be given to a second group of employees who utilize those specifications to write a new program.

This comment addresses the legal concerns created by reverse


9. One court has explained the reverse engineering of computer programs by the following:

Computer programs are written in specialized alphanumeric languages, or "source code." In order to operate a computer, source code must be translated into computer readable form, or "object code." Object code uses only two symbols, 0 and 1, in combinations which represent the alphanumeric characters of the source code. A program written in source code is translated into object code using a computer program called an "assembler" or "compiler," and then imprinted onto a silicon chip for commercial distribution. Devices called "disassemblers" or "decompilers" can reverse this process by "reading" the electronic signals for "0" and "1" that are produced while the program is being run, storing the resulting object code in computer memory, and translating the object code into source code. Both assembly and disassembly devices are commercially available, and both types of devices are widely used within the software industry.


Another way to decompile a program is by "peeling" a silicon chip after the program is imprinted. The layers of the silicon chip are peeled back and analyzed under microscope, one at a time. From these peeled-back layers the object code is discovered, but each "1" and "0" must be deciphered one bit at a time. This type of decompilation was at issue in Atari Games Corp. v. Nintendo of America, Inc., 24 U.S.P.Q.2D 1016, 1017-1018 (CAFC 1992).


engineering of software. Specifically addressed are the legal concerns and protections of companies which write original programs, and the methods useful in preventing reverse engineering. The aspects of a computer program that a company which has a reverse engineering policy can legally use from another's program are also addressed. Finally, given the vagueness of this area of copyright law, the author proposes that specific legislation should be enacted to deal with the complexity of issues involving the legal protection of computer programs.

II. BACKGROUND

The Constitution granted Congress the power to create copyright protection and confer a limited monopoly on the author's creation.\(^{12}\) This protection is given in exchange for the author's disclosure of his work to the world, which in theory will ultimately benefit the public in general.

The goal of copyright law is the advancement of the public welfare by encouraging development and disclosure.\(^{13}\) In exchange for this disclosure to the public, the author is given a monopoly over their work for their lifetime plus 50 years.\(^{14}\) "Congress has granted copyright monopolies to serve the public welfare by encouraging authors to generate new ideas and disclose them to the public, being free to do so in any uniquely expressed way they may choose."\(^{15}\) In regards to computer programs the monopoly should be given very carefully,

Drawing the line too liberally in favor of copyright protection would bestow strong monopolies over specific applications upon the first to write programs performing those applications and would thereby inhibit other creators from developing improved products. Drawing the line too conservatively would allow programmer's efforts to be copied easily, thus discouraging the creation of all but modest incremental advances.\(^{16}\)

---

12. U.S. CONST., art. I, § 8, clause 8: "The Congress shall have the power to . . . promote the Progress of Science and useful Arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries."

13. See Feist Publications v. Rural Telephone Service Co., 111 S.Ct. 1282, 1290 (1991). "The primary objective of copyright is not to reward the labor of authors, but "to promote the Progress of Science and useful Arts."


However, as the Supreme Court has said,

The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance the public welfare through the talents of authors and inventors in science and useful arts.\(^{17}\)

In 1976 Congress revised the copyright law so that all original works are covered by federal law and state common law protection would be preempted.\(^{18}\) Copyrightable subject matter is defined in 17 U.S.C. § 102 as "original works of authorship fixed in any tangible medium of expression." Section 102(b) serves to limit copyright protection to expressions, not ideas or processes.\(^{19}\) The *Lotus*\(^{20}\) court explained it in this way: "The interplay between sections 102(a) and 102(b), illumined by the related legislative history, manifests that the statute extends copyright protection to expressive elements of computer programs, but not to the ideas, processes, and methods embodied in computer programs."\(^{21}\) Original works of authorship refer to works that have been independently created by an author, regardless of their literary or aesthetic merit, ingenuity, or qualitative value.\(^{22}\)

A. **CONTU and the Addition of § 117**

In 1980, Congress following the advice of the National Commission on New Technological Uses of Copyrighted Works (CONTU), added section 117 to the Copyright Act.\(^{23}\) Under copy-

---

18. 17 U.S.C. § 301(a) "...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."
19. 17 U.S.C. § 102(b) "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."
21. Id. at 53.
23. 17 U.S.C. § 117 reads as follows:

Notwithstanding the provisions of section 106, it is not infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or
(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.
right law, computer programs are presently included in the category of literary works\(^{24}\) and are defined as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."\(^{25}\)

**B. Extension of Copyright Protection to Computer Programs**

As indicated above, copyright protection has been extended to computer programs, both object code\(^{26}\) and source code,\(^{27}\) and also to operating system programs.\(^{28}\) In *Apple v. Franklin*\(^{29}\) the court said, "The legislative history of § 102(b) was intended to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law."\(^{30}\)

Copyright law does not give protection to the idea of a computer program, just the expression of that idea in the program.\(^{31}\) This doctrine, which was ultimately codified as § 102(b), originated in the case of *Baker v. Selden*.\(^{32}\) In this famous case, the U.S. Supreme Court ruled that the plaintiff could copyright the expression of his accounting method which was embodied in his book, but copyright protection would not be extended to the actual accounting method itself.\(^{33}\) "Congress chose to extend copyright protection to original expression embodied in computer programs, but not to any idea, method, or process described by the expression."\(^{34}\) Thus, the underlying idea of a computer program will not be afforded

---

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

---

24. Literary works are defined in § 101 as:

- works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied. 17 U.S.C. § 101 (1988).


26. Object code uses only two symbols, 0 and 1, in combinations which represent alphanumeric characters. *See supra* note 9.

27. Computer programs are written in specialized alphanumeric languages, or "source code." *See supra* note 9.


29. *Id.* at 1252.

30. *Id.* at 1253, quoting H.R. REP. No. 1476 at 57.


33. *Id.*

copyright protection, but the original expression of that idea will be.35

Copyright law will also not protect useful articles36 or the utilitarian aspects of such articles. Computer programs could be considered useful articles because they do have an "intrinsic utilitarian function."37 But "elements of expression, even if embodied in useful articles, are copyrightable if capable of identification and recognition independently of the functional ideas that make the article useful."38 Furthermore "[i]f, however, the expression of an idea has elements that go beyond the obvious, and if there are numerous other ways of expressing the non-copyrightable idea, then those elements of expression, if original and substantial, are copyrightable."39

Copyright protection begins once a work is created.40 To be granted copyright protection, a work must also be fixed in a tangible medium of expression.41 While it is true that ownership and copyright protection vest in the owner once the work is fixed, the work must also be registered as a copyrightable work with the copyright office before an infringement action can be filed.42

Because computer programs are included in the category of literary works, when determining the applicable case law to apply, a court does not have to be confined to decisions in which there has been alleged copying of a computer program. Many non-technical decisions have issues which can be applied to computer programs. For example, in the very famous case of Nichols v. Universal Pic-

35. Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986). This case involved a computer program designed to aid in the business of a dental laboratory. The infringer had taken this program and adapted it so that it could be used on another computer. The court in this case decided that, "the idea is the efficient organization of a dental laboratory," anything else was considered to be expression, such as the structure of the program, and thus was protectible by copyright law. Id.

36. 17 U.S.C. § 101 defines useful articles as "an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information. An article that is normally a part of a useful article is considered a useful article."

37. Id.


39. Id.

40. 17 U.S.C. § 302(a) "Copyright in a work created on or after January 1, 1978, subsists from its creation . . . ."

41. 17 U.S.C. § 101 "A work is 'fixed' in a tangible medium of expression when its embodiment in a copy . . . is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."

42. 17 U.S.C. § 411(a) "no action for infringement of the copyright in any work shall be instituted until registration of the copyright claim has been made in accordance with this title."
tures Corp., Judge Learned Hand said, "Even if an infringer does not copy the words or dialogue of a book or play, or the score of a musical work, infringement may be found if there is copying of the work's expression of setting, characters, or plot with a resulting substantial similarity." This same general idea can also be applied to infringement cases involving computer programs.

C. Copyright Infringement

To establish a prima facie case of copyright infringement a plaintiff must prove two elements: 1) ownership of a valid copyright, and 2) copying of constituent elements of the work that are original. Ownership of a valid copyright in a computer program is proven by establishing the originality and copyrightability of the program, as well as compliance with statutory formalities, including registering the work with the Copyright Office. The issued certificate of registration will then give the author a presumption of validity in an infringement action.

To be eligible for copyright protection, a work must be "original" and "fixed in some tangible medium of expression." A computer program is fixed once it is stored onto a disk or written down on a piece of paper. A work is original according to the Supreme Court if "the work was independently created by the author, and it possesses at least some minimal degree of creativity."

The second element of infringement, copying, is sometimes difficult to prove without an admission by the allegedly infringing party. Since direct copying is very difficult to prove, this element can be shown by circumstantial evidence of access to the copyrighted work and substantial similarity between that work and the allegedly infringing work. In Sid & Marty Krofft Television v. McDonald's Corp., the court described two tests that can be used

44. Id. at 121.
46. Sid & Marty Krofft Television Productions Inc. v. McDonald's Corp., 562 F.2d 1157, 1162 (9th Cir. 1977).
48. 17 U.S.C. § 410(c) "The certificate of a registration made before or within five years after first publication of the work shall constitute prima facie evidence of the validity of the copyright . . . ."
52. Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corporation, 562 F.2d 1157, 1164 (9th Cir. 1977).
53. Id.
to prove substantial similarity. The first of these tests labeled "extrinsic," asks the question of whether or not there is a similarity of ideas between the two works at issue. The second test, labeled "intrinsic," looks at whether there are substantial similarities between the forms of expression used in the competing works. Both the intrinsic and the extrinsic test are to be determined by the trier of fact. If the trier of fact first determines that there are substantial similarities in ideas, the trier of fact "must decide whether there is substantial similarity in the expression of the ideas so as to constitute infringement."

Substantial similarity can be found between the actual programs; between the structure of the programs; or as a similarity in the "look and feel" of the nonliteral expressions of the computer programs. In Whelan the court stated that the concern should be whether there are "overall similarities between the programs" and "whether the most significant steps of the programs are similar." In this case, the court concentrated not on the quantity of copying but rather on the quality of the items that were copied.

D. Protection of Computer Programs By Patent

Computer programs can also potentially be protected under the patent laws. Patentable subject matter is defined as "any new and useful process, machine, manufacture, or composition of matter." Usually, patented computer programs are deemed patentable subject matter as "processes." A process is "a mode of treatment of certain materials to produce a given result. It is an act, or a series of acts, performed upon the subject matter to be transformed and reduced to a different state or thing."

A computer program will not be given patent protection if it is

---

54. Id. at 1164.
55. Id.
56. Id.
60. Whelan, 797 F.2d at 1222.
61. Id. at 1246.
62. Id. at 1245. "Because we are concerned with the overall similarities between the programs, we must ask whether the most significant steps of the programs are similar."
determined to be merely a mathematical algorithm. A reason for
this policy is that patent protection awards a monopoly for seven-
teen years, and this protection should not be given to mathemati-
cal algorithms or formulas, otherwise the patentee could preclude
others from using these formulas for the statutory time period. The
Supreme Court stated the rule this way, “Phenomena of nature,
though just discovered, mental processes, and abstract intellectual
concepts are not patentable, as they are the basic tools of scientific
and technological work.”

In the case of In re Iwahashi, the applicant had been denied a
patent for his auto-correlation unit, a device used to obtain a neces-
sary coefficient by taking the square of the sum of two factors in the
equation instead of using multiplication which was more expensive.
The court in this case overruled the examiner and awarded the ap-
plicant a patent stating, “It is no ground for holding a claim is di-
rected to nonstatutory subject matter to say it includes or is directed
to an algorithm. This is why the proscription against patenting has
been limited to mathematical algorithms and abstract mathematical
formulæ which, like the laws of nature, are not patentable subject
matter.”

If the program uses an algorithm in its method, it will be neces-
sary to determine if the algorithm is merely a mathematical formula
or scientific truth. If the algorithm is not a mathematical formula
or scientific truth, and it somehow transforms the subject matter to
a different state or thing, it is patentable subject matter. The court
in Iwahashi used the “Freeman-Walter test” to determine if a
claim defines nonstatutory subject matter. The court stated the test
this way:

Determination of whether a claim preempts nonstatutory subject
matter as a whole, in the light of Benson, requires a two step
analysis. First, it must be determined whether the claim directly
or indirectly recites an “algorithm” in the Benson sense of that
term, for a claim which fails even to recite an algorithm clearly
cannot wholly preempt an algorithm. Second, the claim must be

that can be conducted in old computers, like mental processes and abstract intellectual con-
cepts, are not patentable processes within the meaning of § 101.”
70. Id. at 1374.
the list of non-patentable subject matter not within the scope of section 101.”
72. Iwahashi, 888 F.2d at 1374.
further analyzed to ascertain whether in its entirety it wholly preempts that algorithm.\textsuperscript{73}

Apart from algorithms, it is now also possible to obtain a patent on a graphical user interface. For example, a patent entitled "A System and Method for Managing Graphic Images" was recently issued to Hullot \textit{et al.} and assigned to NeXT Computer, Inc.\textsuperscript{74} The abstract of this patent reads:

A graphic user interface for a computer is provided in which representations of application programs can be placed on the display in a specified area reserved for such a purpose in which area they could not be so readily obscured and forgotten, and which includes a facility for controlling the placement of such representations within the reserved area. The graphic images are guided into specific locations, or "docks" in the reserved area, and their removal from the docks is restricted to prevent accidental withdrawal.\textsuperscript{75}

Because the validity of this type of patent has yet to be tested in court, companies which invest in this type of patent do so at their own risk and should realize such patents may be invalidated in infringement litigation.

Also, unless a computer program is somehow working in connection with some kind of hardware, then patent protection is not applicable. Because of this limitation, the majority of computer programs will find optimal protection from infringement in the copyright laws.

\section*{III. The Clean Room}

Companies use a "clean room" in order to make the proof of the copying element more difficult for plaintiffs to prove in infringement cases. The "clean room" serves to insulate the first group of employees who reverse engineer the competitor's program, from the second group of employees who then use the information acquired from the first group to write a similar program.

A step-by-step approach is useful to analyze exactly how a "clean room" situation works. This example involves a hypothetical situation where one company (Company A) desires to achieve compatibility with another company's (Company B) program. First, a group of employees from Company A analyzes Company B's program. Their task is to reverse engineer or take apart the

\textsuperscript{73} Id.
\textsuperscript{74} U.S. Patent No. 5,146,556 issued on Sept. 8, 1992.
\textsuperscript{75} Id.
program to learn the method, structure, and specifications of Company B’s program. Once this is done, that information is turned over to a second group of Company A employees. The task of this second group is to take the information acquired by the first group and use it to write a new program that will be compatible with Company B’s original program. Copying in this case is difficult to prove, because the second group of employees never actually had access to the original program, they only had access to the information that the first group of employees learned and recorded from Company B’s program. So, when they wrote the new program, it was not copied from the original, but only utilized the specifications from that program.

There are some who think that the very act of decompiling a program could be considered unlawful. Section 117 of the Copyright Act allows the owner of a software program to make a copy so long as “such a new copy of adaptation is created as an essential step in the utilization of the computer program . . . or that such new copy or adaptation is for archival purposes only.” However, there is a split in the circuits regarding the meaning of the statutory language. At least one court has declared that it is legally permissible for a copy to be made in order to reverse engineer a program, while another has stated that this type of copying is not permitted by Section 117.

Most copyright cases do not stop at section 117. After copying is found it is also necessary to determine if the programs at issue are indeed substantially similar. One court has extended copyright protection in computer programs “beyond the programs’ literal code to their structure, sequence, and organization.” The question of substantially similarity is one of fact. The general test is whether an average lay observer would recognize the alleged copy

77. In Vault Corp. v. Quaid Ltd., 847 F.2d 255 (1988), the court stated that “[S]ection 117(1) contains no language to suggest that the copy it permits must be employed for a use intended by the copyright owner, and, absent clear congressional guidance to the contrary, we refuse to read such limiting language into this exception.”
78. In Hubco Data Products Corp. v. Management Assistance Inc., 219 U.S.P.Q. 450, 456 (D. Idaho 1983), the court stated “this expanded definition makes clear that the input of a work into a computer results in the making of a copy, and hence that such unauthorized input infringes the copyright owner’s reproduction right.”
79. Because direct copying is difficult to prove, courts have allowed this element of the prima facie case of infringement to be proven by showing that the defendant had access to and there is substantial similarity between the competing works. See Sid & Marty Krofft Television Productions, Inc. v. McDonald’s Corp., 562 F.2d 1157 (9th Cir. 1977).
as having been appropriated from the copyrighted work.\textsuperscript{81}

IV. CASE DECISIONS INVOLVING REVERSE ENGINEERING

A. E.F. Johnson Co. v. Uniden Corp. of America

Reverse engineering of a program by employees of a company does not automatically mean that they have violated the copyright laws. It is what that company does with the code that determines if there has been copyright infringement. In \textit{E.F. Johnson Co. v. Uniden Corp. of America}, the court stated it this way:

The mere fact that defendant's engineers dumped, flow charted, and analyzed plaintiff's code does not, in and of itself, establish pirating. As both parties' witnesses admitted, dumping and analyzing competitor's codes is a standard practice in the industry. Had Uniden contented itself with surveying the general outline of the EFJ program, thereafter converting the scheme into detailed code through its own imagination, creativity, and independent thought, a claim of infringement would not have arisen.\textsuperscript{82}

In this case, the plaintiff, Johnson, had developed a logic trunked radio system program of mobile radios.\textsuperscript{83} In order to make a compatible radio, the defendant, Uniden, disassembled and copied the program to use with its radios, even going so far as copying the errors and unnecessary information in the program.\textsuperscript{84} Direct evidence of copying was not available to the plaintiff but was inferred from proof of access and substantial similarity.\textsuperscript{85} Johnson was aided in the proof of infringement by the fact that the same errors and unnecessary information appeared in both programs, "The existence of the identical unnecessary instructions in both codes is strong proof of substantial similarity."\textsuperscript{86} The presence of identical errors in copyrighted and infringing computer programs has also been held to be evidence of copying in other cases as well.\textsuperscript{87}

\textsuperscript{81} E.F. Johnson Co. v. Uniden Corp. of America, 623 F.Supp. 1485, 1492 (D.C. Minn. 1985).
\textsuperscript{82} Id. at 1501 n.17.
\textsuperscript{83} Id. at 1488.
\textsuperscript{84} Id. at 1495.
\textsuperscript{85} Id. at 1492.
\textsuperscript{86} Johnson, 623 F.Supp. at 1496.
\textsuperscript{87} Williams Electronics, Inc. v. Arctic International, Inc., 685 F.2d 870, 876 (3d Cir. 1982).
B. **Hubco Data Products Corp. v. Management Assistance Incorporated**

In *Hubco Data Products Corp. v. Management Assistance Incorporated*, Management Assistance Incorporated (MAI) designed and copyrighted an operating system program which allowed access to memory on its memory boards. MAI would place governors on these memory boards to restrict access to the memory in some cases, in order to provide a less expensive system. Hubco developed "The Nilsson Method II," which disassembled, located, and removed the governors on MAI's system, allowing the customer to upgrade. The court in this case said that Hubco could have independently developed, manufactured, and sold its own object code and operating system legally, but that it was infringement for Hubco to copy MAI object codes and sell them in the form of operating systems to MAI computer owners.

C. **Sega Enterprises, Ltd. v. Accolade, Inc.**

In this recent Ninth Circuit case the issues centered on compatibility between video games. Accolade copied a Sega video game to obtain compatibility with the Sega Genesis game system. Accolade decompiled the machine-readable object code from a Sega game in order to achieve compatibility with the Sega system for games that it wished to independently create and market. Accolade then created a manual containing only the functional specifications of this decompiled code and not any of Sega's actual code. Accolade next created its own games for the Sega Genesis system using only the functional specifications. The first issue in the case was whether or not this intermediate copying by Accolade constituted copyright infringement or was it a fair use of the code.

Another issue the court decided was whether a screen display of Sega’s trademark by the Accolade games was a Lanham Trade-

---

89. *Id.* at 452.
90. *Id*.
91. *Id*.
92. *Id.* at 455.
94. *Id.* at 1563.
95. *Id*.
96. *Id*.
97. *Id*.
Sega included a trademark security system which was required on the game cartridge in order for the system to recognize the particular game. Accolade found this code during its reverse engineering of the Sega game and added it to their manual as a standard header to be included in all games. Thus, when Accolade games are inserted into the Sega Genesis system, the console reads this trademark security system initialization and the Sega trademark is flashed up on the screen. The district court ruled that this trademark security system code was not functional and Accolade could not use such a defense to the trademark infringement claim. The district court also ruled in favor of Sega on the copyright claim and issued an injunction against Accolade, also requiring the recall of all of Accolade's infringing games. The Ninth Circuit then stayed the injunction and recall order, and decided to hear the case.

As to the trademark issue, the appeals court ruled in Accolade's favor stating that "when there is no other method of access to the computer that is known or readily available to rival cartridge manufacturers, the use of the initialization code by a rival does not violate the [Lanham] Act even though that use triggers a misleading trademark display." The appeals court also ruled in favor of Accolade on the copyright issue ruling that "when the person seeking the understanding has legitimate reason for doing so and when no other means of access to the unprotected elements exists, such disassembly is as a matter of law a fair use of the copyright work." Accolade raised four arguments in its defense, but fair use was the only one accepted by the court.

---

99. Id.
100. Id. at 1564.
101. Id.
102. Id.
104. Id.
105. Id.
106. Id. at 1562.
107. Id.

\[\ldots\] In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include:

(1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
(2) the nature of the copyrighted work;
(3) the purpose and character of the use in relation to the copyrighted work as a whole; and
Analyzing the first fair use factor, "the purpose and character of the use," the appeals court observed that the fact that Accolade copied for a commercial use weighs against a finding of fair use. But, because the copying by Accolade was only at an intermediate level the court decided that any "commercial 'exploitation' was indirect or derivative" and of "minimal significance. The court ruled that this first factor weighed in favor of Accolade.

In its analysis of the second fair use factor, "the nature of the copyrighted work," the court notes that not all copyrighted works are entitled to the same degree of protection, and no protection extends to the functional or factual aspects of a work. The court stated that "computer programs are, in essence, utilitarian articles . . . . they contain many logical, structural, and visual display elements that are dictated by the function to be performed, by considerations of efficiency, or by external factors such as compatibility requirements and industry demands." The court further notes that if Accolade was to understand the functional requirements of the Sega system, then disassembly of the object code was necessary. Because the Sega video games contain functional aspects which are unprotected by copyright law and cannot be understood without copying, they are afforded a lower degree of protection than other traditional literary works. The court then decided that the second factor also weighed in favor of Accolade.

The court decided that the third fair use factor, the purpose and character of the use, weighed against Accolade because they copied and disassembled the entire game. The fact that Accolade did copy and disassemble the entire game did not preclude an ultimate finding of fair use however, because the ultimate, as opposed to the direct, use by Accolade was limited and therefore the court put very little weight on this third factor.

Finally, the court decided that the fourth fair use factor, the effect on the potential market for the copyrighted work, weighed in favor of Accolade because any loss Sega would suffer would be a

---

110. Id. at 1569-1570.
111. Id. at 1571.
112. Id.
113. Id. at 1572-1573.
115. Id.
116. Id.
minor economic loss. The court put great weight on the fact that if competitors like Accolade were not able to develop their own games and make them compatible, then Sega would enjoy a monopoly of the market and such a monopoly would "run counter to the statutory purpose of promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine." The *Sega* court then put these four fair use factors together and decided that as a matter of law, Accolade was entitled to a fair use defense to its copying. In its summary of this issue, the court stated that "[u]nder the Copyright Act, if a work is largely functional, it receives only weak protection. 'This result is neither unfair nor unfortunate. It is the means by which copyright advances the progress of science and art.' "

V. APPLICATION OF THE LAW

For a company desiring compatibility with another company's hardware or software, the law is clear. The desire to achieve compatibility or standardization does not take precedence over the rights of the author's monopoly in dissemination of their work. For example in *Apple v. Franklin,* the defendant, Franklin, copied Apple's operating system in order to achieve compatibility with the Apple II system. In this case the court said, "If other programs can be written or created which perform the same function as Apple's operating system program, then that program is an expression of the idea and hence copyrightable." A company desiring to achieve compatibility can do this legally by anything but direct copying.

A. What can a company, which develops and manufactures software do to protect its programs from being copied?

A company should first take steps outside of the legal realm to

117. *Id.* at 1570-1571.
118. *Id.* at 1571.
123. *Id.* at 1243.
124. *Id.* at 1253.
125. "Franklin may wish to achieve total compatibility with independently developed application programs written for the Apple II." *Apple v. Franklin,* 714 F.2d at 1253.
prevent others from reverse engineering its programs. As many safeguards as possible should be implemented within the program in order to protect against the disassembly and copying of the program. But, "hackers" have become very creative and there are few programs that cannot be disassembled.126

Next, some legal action may be taken to deter would-be "COPYERS." First, a copyright notice should be placed on the disk that contains the program, on the package in which the disk is sold, in the software manual, and most importantly, copyright notice should be placed in the actual code of the program.127 Extraneous and unneeded instructions and subroutines should be hidden in the code very carefully. The idea is to make them appear like useful instructions to the casual observer. If they show up in the alleged copy they will be prima facie evidence of copying,128 because it is highly unlikely that two independently created programs would contain the same errors or unneeded instructions. This also makes the would-be copier's task more difficult if they try to search for unnecessary code or subroutines. If the would-be copier has to analyze every line of computer code and determine its function in the program, the process becomes very time consuming and expensive. Finally, a company should aggressively enforce its copyrights in court when it becomes evident that their work product has been directly copied by a competitor.

B. What can a company which practices disassembly do to protect itself?

A company which regularly disassembles programs may also protect itself from litigation. The key is to utilize and incorporate as much independent creation as possible into the new program. Anything that has been independently created, although it may be based on someone else's idea will not constitute infringement. It is important to separate the idea and function of a program from the expression of its code. Ideas and functions may be used to create a

126. See Vault Corp. v. Quaid Software Ltd., 847 F.2d 255 (5th Cir. 1988) "Vault had a disklock program which prevented unauthorized copying of that disk. Quaid invented a key to that program which unlocked the disk and allowed the user to copy it."

127. 17 U.S.C. § 401 "If a notice of copyright in the form and position specified by this section appears on the published copy or copies to which a defendant in a copyright infringement suit had access, then no weight shall be given to such a defendant's interposition of a defense based on innocent infringement in mitigation of actual or statutory damages. . . ."

new program and user interface, but in no case should actual code ever be copied from a competitor's program.

Companies which routinely analyze programs produced by others should do everything possible to separate its product from the original program. If for some reason a company desires to achieve compatibility with another's product, and this cannot be done by independent creation, then that company should seek a licensing agreement from the owner of the original program. In some cases this may be expensive, but avoidance of court costs alone might make these agreements more economically justifiable. This is particularly true in the situation where costs would be incurred in the development of a program that might be ruled a copy.

VI. SOLUTION AND PROPOSAL

At the present time, authors of computer programs may obtain legal protection for their work in the areas of patent, copyright, trade secret, trademark, unfair competition and arguably trade dress. While this may be acceptable to some, this author and others view such provisions as cumbersome and in need of replacement. Often plaintiffs in infringement suits are forced to plead many different causes of action in order to ensure recovery for what is in essence, copying by the defendant. Also the 75 year duration of copyright protection is too long for computer technology. The software industry has exploded in the last ten to fifteen years and many advances have been made which make old technologies useless. While the useful lifetime of a computer program is not known at this time, it is known that technology is changing so rapidly that a monopoly for 75 years is commercially an eternity. Upgrades and new programs for old applications are introduced daily due to technological advances and the resultant improvement in software programs.

129. Id. at 1103. "The unique characteristics of computer software — its ease of copying; its various aspects (algorithms, source code, object code, and user interface); and its simultaneously functional and literary nature — have made the application of traditional intellectual property principles to software difficult." Id.

130. For corporations, 17 U.S.C. § 302(c) provides: "In the case of . . . a work made for hire, the copyright endures for a term of seventy-five years from the year of its first publication, or a term of one hundred years from the year of its creation, whichever expires first." 17 U.S.C. § 302(c).

For an individual authors, protection is provided for life plus 50 years. 17 U.S.C. § 302.

131. "In the high technology area of computers . . . the economic life of an innovation may only be a few years." Andrew G. Rodau, Protecting Computer Software: After Apple Computer, Inc. v. Franklin Computer Corp., Does Copyright Provide the Best Protection?, 57 TEMPLE L.Q. 527, 532 (1984).
While the 17-year monopoly provided by patent is still too long for computer software, the alternative of trade secret protection is virtually limitless, as long as the invention can be kept secret. Reliance on these traditional legal theories has detracted from the competitive interaction which made the software industry what it is today.133

Congress should establish a workable compromise by enacting new legislation which deals exclusively with computer software and the issues concerning protection and infringement of computer software. This legislation should extract elements from each of the areas mentioned in the above paragraphs as well as elements from other areas of the law including contracts and torts. The Semiconductor Chip Act134 was enacted to protect mask works through basically the same mechanism, and could be used as a model.

The goals of this legislation should be to:

1. Foster and maintain the competitive edge that has been achieved by the U.S. Computer Software Industry.135
2. Reward computer software inventors for their time and financial investments.136
3. Promote the development of new software and improvements of old software.
4. Encourage the free dissemination of ideas embodied in popular and innovative software programs.137
5. Allow others to use elements of programs that have come to be standards in the industry.
6. Provide meaningful penalties for violation of this legislation.

As mentioned above, the U.S. software industry is viewed by many as the best in the world.138 Any new legislation to protect property interests in computer programs should first serve to protect and enhance the position already achieved by American

132. 35 U.S.C. § 154 "Every patent shall contain a short title of the invention and a grant to the patentee, his heirs or assigns, for the term of seventeen years."
135. See Toole, supra, note 4.
136. Copyright does not extend protection for the actual work done by an inventor. In Feist Publications v. Rural Telephone Service Co., 111 S.Ct. 1282, 1295 (1991) the Court stated, "the 1976 revisions to the Copyright Act leave no doubt that originality, not "sweat of the brow," is the touchstone of copyright protection in directories and other fact-based works."
137. "New ideas must be disseminated throughout society so that further progress can be made without having to reinvent the wheel." Brown, supra note 133 at 978.
138. See Toole, supra note 4.
Software reverse engineering is needed to encourage new development and innovation in computer software. Thus, the open exchange of ideas and techniques should be balanced with the need to reward inventors. There also should be encouragement for other companies to use elements of programs that have developed into industry standards. Traditionally, American society has allowed the first market entrant to set the industry standard. But, the manner by which some software companies are presently utilizing copyright protection allows the first market entrant to control the market. "The first developer of successful software who is able to set an industry standard may now use the copyright law as a shield against competitive market forces."

Software companies are now choosing to actively enforce property interests in their programs through copyright law. This enforcement could have an adverse effect on innovation as development of new products may decline and standardization could become impossible. Society can only benefit from the dissemination of innovations developed by software developers, whether this is accomplished by reverse engineering or licensing agreements. Any new legislation to protect property interests in computer software should promote standardization of computer programs as much as possible. "Standardization is particularly important for computer software, where compatibility is essential for the sharing of data between programs."

The original developer should however, receive some benefit from another's use of his program. Any new legislation should include economic incentives to entice developers to continually create new innovations. This should be balanced with the open dissemination and teaching necessary to promote standardization and efficiency, thus minimizing duplication of effort. This author suggests a system modeled after the music industry's compulsory licenses. Such protection serves to reward the first innovator who brings

139. An industry standard would be an element of a program that is accepted as such by the market. Examples include the user interfaces of certain programs.
140. Examples of this are the convention of having the brake pedal to the left of the gas pedal and the "figure-H" pattern of an automobile stick. Brown, supra note 133 at 977.
141. Id.
142. See Kapor, supra note 7.
143. Toole, supra note 4.
144. Id. at 147.
145. Id. at 150.
146. 17 U.S.C. § 115(a)(1) "When phonorecords of a nondramatic musical work have been distributed to the public in the United States under the authority of the copyright
something new to the market but also makes this technology readily available to others who wish to adapt it for use in their own programs. Exact copying would not be permitted unless authorized by the owner of the technology. But, a new method or interface could be used by others' programs, as long as a per unit license fee was paid to the developer of that method or interface. As within the music industry's compulsory license, the license fee could be set by statute. Or, a lesser fee could be negotiated between the user and the owner.\textsuperscript{147} The original author of the software program could be given a shortened statutory period such as two years in which to exploit the exclusive right to use the parts of the program. After this two year period, the original author would still own the copyright in the program, but others would be free to use it upon payment of the compulsory license fee or negotiation of a direct license with the original author.

One potential problem with such legislation, is that some developments could be viewed as trivial, and therefore should not be governed by compulsory licenses. A way around this problem is to adopt a standard much like the patent statute's nonobvious standard\textsuperscript{148} and require any potential licensor to first register any innovative method or interface before becoming eligible to collect fees from any company taking advantage of the compulsory license statute. This statute should also include an element which requires the original developer to disclose any innovative methods or interfaces for which she wishes to receive compulsory fees. This still allows the developer to protect the technology through trade secret if that would further the company's best interests. Of course, any technology protected by trade secret should lend itself to secrecy and meet the statutory guidelines for trade secrets.\textsuperscript{149}

Under this proposed system, because one company should not be able to exclude all others, the computer software industry should become more efficient. Innovation will be encouraged and rewarded, but perhaps even more important is that improvement of owner, any other person may, by complying with the provisions of this section, obtain a compulsory license to make and distribute phonorecords of the work."

\textsuperscript{147} "The usual effect of the system is to make the statutory royalty rate a ceiling on the price copyright owners can charge for use of their songs under negotiated contracts: if the owner demands a higher price in voluntary negotiations, the manufacturer can turn to the statutory scheme, but if the owner is willing to accept less than the statutory rate, he is free to do so." Recording Industry Association of America v. Copyright Royalty Tribunal 662 F.2d 1 (D.C. Cir. 1981).


\textsuperscript{149} California Civil Code § 3426.1(d).
existing programs will be allowed and even encouraged.\textsuperscript{150}

VII. CONCLUSION

Computer programs can now be protected under various theories of intellectual property law. As long as these traditional theories are in force, software developers should be aware that there are some activities which are prohibited when reverse engineering or analyzing another company's program. No actual code should ever be copied; all of the code for a new program which is based on another's program, should be written independently. Companies trying to protect their programs from reverse engineering should incorporate as many safeguards as possible into the programs. Such safeguards include devices which prevent copying, and the purposeful inclusion of unneeded code in the program which will assist the company in meeting its proof burden should it be necessary to prove instances of copying during infringement litigation.

New legislation should be enacted to better protect computer programs and the American software industry. One potential model is provided by the compulsory licenses now in use by the music industry. This method of protection would allow free dissemination of the ideas and methods used in innovative software development and should serve to enhance the overall efficiency of the software industry. While such change may initially be difficult to implement, it is long overdue and new legislation should be enacted before it is too late.

\textsuperscript{150} "But, it seems likely that more programmers will take advantage of opportunities to improve an existing application by making it faster, easier to use, or more functional." Whitmeyer, \textit{supra} note 128 at p. 1120.