

ARTICLE

INTELLECTUAL PROPERTY PROTECTION AND REVERSE ENGINEERING OF COMPUTER PROGRAMS in the united states and the european community*Charles R. MCManis* †**Table of Contents**

I. INTRODUCTION 26

II. reverse engineering 28

III. forms of intellectual property protection for computer programs in the united states 31

A. Trade Secret Protection for Publicly Distributed Computer Programs 32

B. Intellectual Property Protection for Publicly Distributed Computer Programs 33

C. Computer Program Protection Under United States Copyright Law 39

IV. The status of reverse engineering under United States copyright law and the european community directive 45

A. The *Atari*, *Sega*, and *Vault* Decisions 45B. A Comparison of *Atari*, *Sega*, and *Vault* with the European Community Directive 48

C. Reverse Engineering as a Fair Use Under § 107 50

D. Section 117 Limitations on Exclusive Rights in Computer Programs 76

V. Copyright and Patent Preemption of Restrictions on Reverse Engineering 88

A. Copyright Preemption 88

B. Patent Preemption of State Law and Primacy Over Copyright Law 91

I. INTRODUCTION

Perhaps the single most important intellectual property issue in the United States today is the scope of legal protection to be given computer programs.¹ Legal protection for computer programs can involve as many as four different forms of intellectual property law—trade secret law, copyright law, trademark law, and patent law. When courts have the discretion to decide the precise scope of protection to be provided computer programs under any one of the foregoing bodies of law, the other three must be taken into account, and some reference must be made to a fifth, closely related form of federal intellectual property protection for semiconductor chip designs.² In addition, courts should be aware of developments in foreign and international protection of computer programs, such as the recently promulgated European Community Directive on the Legal Protection of Computer Programs ("EC Directive").³ That a tension exists among these overlapping forms of protection is particularly apparent in cases where a user of a computer program disassembles, decompiles, or in some manner "reverse engineers" the program.⁴

Despite the variety of intellectual property protection available for computer programs, United States courts have only recently dealt in

a comprehensive fashion with the legality of reverse engineering of computer programs. In *Atari Games Corp. v. Nintendo of America*⁵ and *Sega Enterprises v. Accolade, Inc.*,⁶ two separate courts of appeals held that, at least where the computer program involved is part of a lockout device designed to prevent unauthorized electronic game cartridges from operating on a game console, decompilation or disassembly of the program can constitute a fair use under § 107 of the Copyright Act of 1976⁷ ("Copyright Act"). In so holding, the courts explored various points of intersection between copyright protection for computer programs and protections under patent, trademark and semiconductor chip design law.⁸ These two decisions, along with a third, *Vault Corp. v. Quaid Software Ltd.*,⁹ which held that reverse engineering not involving decompilation or disassembly was a permissible use under §117 of the Copyright Act, appear to bring United States copyright law into harmony with two reverse engineering provisions contained in the EC Directive.

The cases also raise two further issues: 1) whether computer software developers who use computer programs as lockout devices to eliminate competition, or who unilaterally place contractual restrictions on the reverse engineering of publicly distributed computer programs, are engaging in copyright misuse, and 2) whether the enforcement of contracts restricting reverse engineering of publicly distributed computer programs undermines and is thus preempted by federal copyright or patent law.¹⁰

Part II of this article provides an overview of the process of, and reasons for, reverse engineering of computer programs.

Part III describes the various forms of intellectual property protection available for computer programs in the United States, with particular emphasis on federal copyright law, and discusses the limitations on each of these forms of protection.

Part IV discusses the extent to which reverse engineering may be permitted under §§ 107 and 117 of the Copyright Act and under the EC Directive. It also discusses whether either use of a computer program as a lockout device to eliminate competition or use of contracts to prevent reverse engineering of publicly distributed computer programs constitutes copyright misuse.

Part V of the article considers the extent to which federal patent law, as well as federal copyright law, might be said to preempt enforcement of contracts that attempt to restrict the right to reverse engineer publicly distributed computer programs.

This article comes to the following conclusions: (1) As under the EC Directive, reverse engineering of publicly distributed computer programs, including decompilation and disassembly, constitutes a "fair use" under § 107 of the Copyright Act, where the reverse engineering is necessary to achieve interoperability with another program; (2) reverse analysis that can be accomplished merely by using a publicly distributed program in conjunction with a computer is a permissible use under § 117 of the Copyright Act; (3) computer software developers who use computer programs as lockout devices to eliminate competition or who unilaterally place contractual restrictions on the reverse engineering of publicly distributed programs may be engaging in copyright misuse; and (4) federal patent law, as well as federal copyright law, should be held to preempt the enforcement of contractual restrictions on the reverse engineering of publicly distributed computer programs.

II. reverse engineering

Computer programs are commonly distributed to consumers only in machine-language (or object-code) form, while the developer maintains the human-readable (or source-code) version of the program¹¹ as a trade secret.¹² The object-code version is virtually impossible for a user to "read" while it remains in object code.¹³ Therefore, users and developers wishing to study the structure and technical parameters of an existing program must gain access to the program in its source-code form.

For this reason, users and developers who lack access to source code may engage in "reverse engineering," which attempts to reverse the steps originally involved in creating a program.¹⁴ Reverse engineering allows a user to create an equivalent of the original source-code version of the program. Methods of reverse engineering range from analyzing screen displays of the object code to decompilation or disassembly of the program.¹⁵

Computer tasks typically require several different "layers" of programs. The top layer is most accessible to (or perhaps even custom-designed by) computer users, and consists of applications programs that interact with the user to perform tasks such as word processing, or creation of a spread sheet or graphics design.¹⁶ These programs are generally embodied in magnetic media, such as tapes or discs; optical media, such as CD-ROMs; or solid-state memories, such as cartridges for games. Beneath that layer is the computer's operating system, which performs basic machine functions such as coordinating application programs, and storing and retrieving information.¹⁷ Operating system programs are generally embodied in magnetic media or solid-state memories. At the base of the computer hierarchy is microcode, a body of instructions embedded in an integrated circuit or semiconductor chip. Microcode transforms the higher-level object-code instructions from the application program or operating system into electronic signals necessary

to control the computer's circuits.¹⁸ Reverse engineering may involve any of these "layers" of computer programs.

Users or developers of computer programs may wish to study the design of a particular program for a variety of purposes. One purpose is to learn or teach new programming techniques, either for purely academic reasons or in order to create a commercially-marketable software product. Another is to make more effective use of the program by identifying and locating program errors ("debugging"), customizing the program to meet the user's own particular requirements, or modifying a program in order to make it compatible with another program or a particular piece of computer hardware ("porting"). A third purpose is to develop a specific, commercially-marketable, non-competing program that will interact compatibly with the reverse engineered program. Here, the developer's goal could either be to create a new application, compatible with an existing operating system, or to improve or counteract features contained in an existing application. Finally, a user may wish to develop a functionally equivalent product that will compete directly with the original program. Such a program may need to "understand" enough of the command and file structure of the reverse-engineered program to interact with the files or documents it produces, or actually employ particular features (e.g., subroutines) it contains.¹⁹

.III. forms of intellectual property protection for computer programs in the united states

Determining the legality of reverse engineering of computer programs is complicated, since computer programs are virtually unique in the field of intellectual property law. Computer programs function both as a part of a machine, in their object-code form, and as a means of communicating with other human beings, in their source-code form, and are thus eligible for federal patent protection as well as federal copyright protection.²⁰ Moreover, computer programs can simultaneously be publicly distributed, in their object-code form, while being kept secret, in their source-code form, and thus may also be eligible for state trade secret protection.²¹ To further complicate matters, some programs can be embedded in semiconductor chips, the designs of which are separately protected by a *sui generis* form of intellectual property protection.²² Screen displays generated by computer programs may be separately protected as copyrightable literary or artistic works, may contain sufficiently distinctive product features to qualify for protection as trademarks, and may even be eligible for federal design patent protection as new and non-obvious ornamental designs for an article of manufacture.²³

A. Trade Secret Protection for Publicly Distributed Computer Programs

1. state trade secret law

The earliest form of legal protection for computer programs was state trade secret law. During the mainframe era of computer hardware development, when the right to use a computer program could be "bundled" with the overall leasing arrangements for mainframe computer hardware, there was little need for any additional legal protection.²⁴ The equipment manufacturer simply negotiated a contract with each user.

Although copyright and patent protection for computer programs have become preeminent in the era of the personal computer and publicly distributed software,²⁵ trade secret law continues to provide an important subsidiary form of protection for source code. To take advantage of state trade secret protection, a developer need only control access to the source code and contractually bind all parties who have access to the source code to maintain its confidentiality.²⁶ State trade secret law not only enforces such provisions, but provides common-law tort remedies for inducing another to breach such provisions or for acquiring another's trade secret by otherwise improper means, such as industrial espionage.²⁷ State trade secret law also prohibits the use of a trade secret by a party who knows or has reason to know that the secret information was improperly acquired or disclosed by a third party.²⁸ Trade secret protection is of potentially indefinite duration, so long as one maintains the secrecy of the information.

However, the scope of trade secret protection is qualified. Trade secret law prohibits the acquisition, use or disclosure of another's trade secret only when it is "improper."²⁹ One is explicitly permitted not only to independently develop trade secrets, but to reverse engineer publicly distributed products.³⁰ Thus, if state trade secret law were the only legal protection available for publicly distributed computer programs, reverse engineering of such programs would clearly be legal.

2. de facto federal protection of trade secrets

The Copyright Office provides a sort of trade secret protection for the source code of copyrighted programs by limiting the circumstances in which works deposited at the Copyright Office can be reproduced.³¹ The Copyright Office also allows those registering computer programs for copyright protection to deposit only the identifying portions of the source-code versions of their programs.³² Under certain circumstances, copyright applicants may block out portions of the source code containing trade secrets.³³

Additionally, under the Copyright Office's "rule of doubt" policy, some copyright applicants may deposit only the object code, along with written assurances that it contains copyrightable authorship.³⁴ These Copyright Office regulations in effect create a federal form of trade secret protection for computer programs.

B. Intellectual Property Protection for Publicly Distributed Computer Programs

1. Copyright, Trademark, and Patent Protection, and the semiconductor chip act of 1984

It is now well established that all computer programs, regardless of their form (source code or object code), function (applications programs or operating system programs), or medium of expression (magnetic tapes or discs, optical media, solid-state memory, or microcoded semiconductor chips), are protectible under § 102(a) of the Copyright Act of 1976 as works of authorship—specifically, as "literary works."³⁵ It is equally well established that screen displays generated by a computer program are separately protectible as works of authorship—either as literary works, or as pictorial, graphic or audiovisual works, depending upon the nature of the particular screen displays.³⁶

Additionally, any aspect of a screen display that either inherently identifies or comes to be associated with the underlying program, and is not otherwise functional, may also be protectible under federal and state law as a trademark. Protection under trademark law applies to distinctive brand names, logos and product features.

Federal patent law also provides protection for publicly distributed computer programs. It is becoming increasingly clear that computer-related inventions that are sufficiently new, useful and non-obvious may be the subject of federal *utility* patent protection, and that ornamental features of a screen display that are sufficiently new, original and non-obvious may be entitled to *design* patent protection. A recent study concluded that the United States Patent and Trademark Office (USPTO) is now issuing significant numbers of "pure" software utility patents.³⁷ A number of such patents have been upheld as valid by federal courts, including the Court of Appeals for the Federal Circuit.³⁸ In addition, the USPTO Board of Patent Appeals and Interferences recently rendered its first decision acknowledging that ornamental elements of computer screen displays may be the subject of design patent protection if they are described, shown and claimed as an integral part of a computer system.³⁹

Finally, as a result of the Semiconductor Chip Protection Act of 1984,⁴⁰ (hereinafter "Chip Act") there is also a new *sui generis* form of federal intellectual property protection for semiconductor chip designs, which the Act refers to as "mask works." The purpose of the Chip Act is to protect the design of semiconductor chips of all kinds, ranging from microprocessors (i.e., "computers on a chip") to mere memory chips designed to store information.⁴¹

2. Limitations on Protection

The protection provided by each of the foregoing bodies of law comes with its own distinct limitations. Federal copyright law protects only an author's original expression, not the ideas expressed, and permits not only free public use of these ideas but also a variety of "fair uses" of otherwise protectible expression.⁴² Factual or utilitarian works, in particular, are protected only against substantial appropriations of literal expression.⁴³ Proof that a work has been independently created, although not sufficient to defeat a claim of patent infringement, will defeat a claim of copyright infringement. On the other hand, federal copyright law requires no great novelty of expression—originality having been defined as requiring little more than that the copyrighted work owe its origin to the author claiming copyright protection and not be a slavish copy of someone else's work.⁴⁴

Federal patent protection for either an invention or a design is limited to subject matter that is not only new, but also non-obvious to those skilled in the art to which the subject matter pertains.⁴⁵ It has been estimated that well over 90 percent of computer programs will fail to meet this requirement.⁴⁶ The issuance of a patent is also conditioned upon the applicant's complete disclosure of the subject matter claimed.⁴⁷ This disclosure allows the public to benefit from the advance in the art, both during and after the expiration of the limited patent term.⁴⁸ Although these disclosure requirements do not necessarily require a patent application for a program-related invention to reveal the literal code of the program involved, there must be sufficient disclosure to ensure that a person of ordinary skill in the art disclosed could practice the invention without undue experimentation.⁴⁹ Where the disclosure requirements are met and a patent is issued, the patent owner may exclude others from (1) making, using or selling the patented invention; (2) importing, using or selling the product of a patented process; or (3) actively inducing or contributing to an infringement,⁵⁰ subject only to certain narrowly tailored "fair" uses.⁵¹

Thus, while the scope of protection is narrower under copyright law than in patent law, the requirements for obtaining copyright protection are more modest. Additionally, the term of copyright protection is considerably longer than that provided for patented

inventions or designs.⁵² Because most computer programs and screen graphics cannot meet the federal patent law requirement of non-obviousness, federal copyright law currently provides the main bulwark of protection in the United States for publicly distributed computer programs.

Other bodies of intellectual property law offer supplemental protection. The federal and state law protecting product features, such as screen graphics, as trademarks provides potentially indefinite protection, but is limited, as federal copyright law is, to protecting "non-functional" features of the product.⁵³ The scope of protection under trademark law is even narrower than that provided under copyright law. Trademark law merely prevents others from using the same or a similar feature in such a way as to create a likelihood of confusion with (or under the statutory law of at least some states a likelihood of "dilution" of the impact of) a distinctive trademark.⁵⁴ Various "collateral uses" of another's mark, such as use in non-deceptive comparative advertising, are considered "fair uses" of the mark.⁵⁵

The federal statute protecting semiconductor chip designs embodied in a semiconductor chip product against unauthorized reproduction provides a ten-year term of protection for such designs.⁵⁶ However, the Chip Act contains a provision explicitly permitting reverse engineering of semiconductor chips.⁵⁷

Cutting across the federal law of patents, copyrights and trademarks is a final limitation: the doctrine of misuse. Where one "misuses" one's patent, copyright, or trademark⁵⁸-that is, exercises one's federal rights in a manner that conflicts with the goals of the statute creating that right, or has an anti-competitive effect within the meaning of federal antitrust law-protection is denied.

Because federal copyright law provides the most significant protection for publicly distributed computer programs, the precise scope of that protection and the effect it has on the legality of reverse engineering warrants separate and more detailed examination.

C. Computer Program Protection Under United States Copyright Law

Although federal copyright protection "subsists" from the moment a work of authorship is fixed in tangible form, copyright protection for computer programs is necessary primarily to prevent publicly distributed, machine-language (object-code) versions of computer programs from being reproduced without authorization of the copyright owner.⁵⁹ The source code version of the computer program, as we have seen, can be, and usually is, protected as a trade secret. Thus, source code has no direct need for the additional protection provided by the 1976 Copyright Act. The Act, however, does play a potentially critical role in the effort by program developers to prevent reverse engineering of computer programs.

Section 106 of the Copyright Act grants a copyright owner the exclusive rights to reproduce and to prepare derivative works based on the copyrighted work.⁶⁰ Federal copyright law would thus appear to prohibit both the unauthorized adaptation of the object-code version of a program, and the unauthorized, even if indirect and not entirely literal, reproduction of the source code.⁶¹ Any attempt to decompile, disassemble or otherwise reverse engineer the object-code version of a program may thus violate federal copyright law.

However, both the subject matter protected under § 102(a) of the 1976 Copyright Act and the exclusive rights granted under § 106 of the Act are subject to limitations. Section 102(b) specifies that copyright protection does not extend to any "idea, procedure, process, system, method or operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in the work."⁶² This is known as the "idea/expression dichotomy."⁶³

A corollary of the idea/expression dichotomy is the "merger doctrine." The merger doctrine states that if there is only one way, or a limited number of ways, of expressing an idea, then the idea and its expression are said to have merged. Copyright protection is unavailable for the expression which has merged with the idea.⁶⁴

Analogous to the idea/expression dichotomy is the fact/expression dichotomy, which severely limits the scope of federal copyright protection under § 106 for factual works.⁶⁵ Moreover, if otherwise protectible expression is inseparable from the facts, ideas or utilitarian functions embodied in the expression, copyright protection is completely unavailable. These long-standing principles not only delimit the subject matter and scope of the exclusive rights protected by the Copyright Act, but also distinguish federal copyright protection from patent protection for computer programs.⁶⁶

The case law applying the Copyright Act to computer programs can be divided into two generations, corresponding to §§ 102 and 106. The first generation was concerned with whether all forms of computer programs were copyrightable subject matter, while the second generation has attempted to define the scope of the exclusive rights held by the owner of a computer-program copyright.

1. First Generation Cases: Computer Programs as Copyrightable Subject Matter

During the first generation of cases concerned with copyright protection for computer programs, a number of commentators argued against granting copyright protection to publicly distributed, machine-readable versions of a computer program.⁶⁷ They argued that providing such protection without requiring that the source code be published would subvert one of the traditional, constitutionally-based, norms of federal copyright law: namely, that bringing new ideas into the public domain is the *quid pro quo* the public receives in exchange for the limited monopoly right the author receives to protect his or her expression of ideas.⁶⁸

Before the advent of computer programs, publicly distributed copyrighted works necessarily communicated the entirety of the ideas in the work. In the case of computer programs, however, "it is possible both to publish a work and keep it secret, and keeping it secret is part of the way the commercial value of the work is maintained."⁶⁹

A further problem posed by computer programs is that the object-code version of a computer program is utilitarian in a more fundamental sense than is virtually any previous class of copyrightable work. Unlike a factual work, a book of instructions, or even a human-readable source-code version of a computer program, a machine-readable, object-code version of a program does not instruct a human being about how a task might be performed. Rather, in conjunction with computer hardware, it performs the task itself.⁷⁰ The principal utility of a machine-readable version of a computer program springs not from the information it conveys to human beings, but from the information it conveys to a machine. Historically, the only federal protection available for works of this sort had been patent protection.⁷¹

Nevertheless, the courts in the first generation of computer copyright cases rejected the argument that object code should be denied copyright protection.⁷² In the second generation of software copyright cases, however, the courts have been more divided over the appropriate scope of protection under § 106 of the Copyright Act. The specific point of contention has been whether and how far copyright protection should extend beyond the literal code to the structure, sequence and organization or other non-literal elements of a program.⁷³

2. Second Generation Cases: The Scope of EXCLUSIVE RIGHTS TO Computer Programs

The second generation of cases, like the first, has its roots in the idea/expression dichotomy. On the one hand, the courts have long recognized that copyright protection "cannot be limited literally to the text, else a plagiarist would escape by immaterial variations."⁷⁴ On the other hand, it is equally clear that copyright law does not prohibit the use of ideas contained in a copyrighted work, but only the copying of protected expression.⁷⁵ Drawing the line between protected expression and unprotected ideas depends in part on whether the copyrighted work is a factual/utilitarian work, in which case non-literal elements of a work are more likely to be classified as unprotected ideas, or a non-factual, imaginative work, in which case more leeway is allowed for classifying non-literal elements as a part of the author's original expression.⁷⁶ Computer programs are difficult to classify because their creation arguably involves a blend of utilitarian and imaginative elements.⁷⁷

Unlike the first generation of computer copyright cases, this second generation of cases has no direct bearing on the question of the permissibility of reverse engineering under the Copyright Act of 1976.⁷⁸ For even if copyright protection for computer programs were limited to prohibiting verbatim reproduction or very close paraphrasing of the program's actual code, as some commentators have argued it should be,⁷⁹ federal copyright law would still seem to prohibit reverse engineering. The whole point of reverse engineering, after all, is to produce a human-readable adaptation of the object code and a virtual reproduction of the original source-code version of the program. Thus, both the indirect reproduction of the source code and the adaptation of the object code may violate the copyright owner's exclusive right, contained in § 106 of the Copyright Act, unless some other provisions of the 1976 Act permit such adaptation and reproduction.

Just as § 102(b) explicitly limits the subject matter protected under § 102(a) of the 1976 Copyright Act, however, so a number of provisions of the 1976 Act limit the scope of the exclusive rights enumerated in § 106.⁸⁰ Two of these provisions arguably permit at least some forms of reverse engineering. First, § 107 permits the "fair use" of another's copyrighted work. Under this section, all forms of reverse engineering, including decompilation and disassembly, may be viewed as a fair means of obtaining access to the unprotected ideas contained in a publicly distributed computer program.⁸¹ Second, § 117(1) permits copying or adapting a computer program as an "essential step" in the utilization of the program in conjunction with a machine.⁸² Provided that the copy is used in no other manner, reverse engineering techniques that take place entirely within the computer may be viewed under this section as a specifically permitted means of utilizing the computer program.⁸³ Each of these provisions of United States copyright law has a counterpart in the recently promulgated European Community Directive on the Legal Protection of Computer Programs.

IV. The status of reverse engineering under United States copyright law and the European Community Directive

A. The Atari, Sega, and Vault Decisions

In *Atari Games Corp. v. Nintendo of America*⁸⁴ and *Sega Enterprises, Ltd. v. Accolade, Inc.*,⁸⁵ two federal courts of appeals for the first time held that § 107 of the Copyright Act permits an individual who is in rightful possession of a copy of a computer program to engage in reverse engineering.⁸⁶ The courts held that such a party may undertake necessary efforts, including disassembly or decompilation of the program, to gain an understanding of the unprotected functional elements of the program, such as the ideas, processes or methods of operation contained in the program, at least where there is a legitimate reason for doing so and no other means of access to the unprotected elements exists.⁸⁷

An earlier, and widely criticized, court of appeals decision, *Vault Corp. v. Quaid Software Ltd.*,⁸⁸ likewise held that loading a copyrighted program into a computer for the purpose of reverse engineering by means other than disassembly or decompilation constituted an essential step in the utilization of the computer program in conjunction with a machine and was thus permitted under § 117 of the 1976 Copyright Act.⁸⁹

In both *Atari* and *Sega*, a game cartridge producer sought to reverse engineer a "lockout" program, used by a leading producer of home video game systems, that was designed to prevent the system's game console from accepting unauthorized game cartridges.⁹⁰ The courts held that reverse engineering was valid for the purpose of creating game cartridges that would be compatible with consoles produced by another company.

Atari had attempted to analyze and replicate Nintendo's "10NES" security system by monitoring the communication between patented computer chips located in the Nintendo console and authorized game cartridges.⁹¹ The effort failed.⁹² Atari next tried to analyze the chips themselves by chemically peeling layers from Nintendo's chips, microscopically examining them, transcribing the object code of Nintendo's lockout program into a handwritten list of ones and zeros, and then keying this transcribed copy of the program into a computer in an attempt to disassemble it.⁹³ Failing again, Atari entered into a licensing agreement with Nintendo that authorized Atari to market compatible game cartridges to Nintendo console owners.⁹⁴

Subsequently, Atari's attorney made misrepresentations to the Copyright Office in order to obtain a copy of the registered source code for Nintendo's 10NES lockout program.⁹⁵ This source-code version of the program, in turn, facilitated a second round of reverse engineering that ultimately enabled Atari to write its own "Rabbit" program to unlock the Nintendo console.⁹⁶

Atari then breached its licensing agreement and sued Nintendo for antitrust violations, unfair competition and an unspecified claim of patent infringement.⁹⁷ Nintendo counterclaimed for, among other things, copyright infringement.⁹⁸ Finding it likely that Nintendo would succeed in establishing its copyright infringement claim, the district court preliminarily enjoined Atari from further exploiting either Nintendo's lockout program or using Atari's own "Rabbit" program.⁹⁹ The court of appeals in *Atari* agreed with the district court that Nintendo was likely to succeed on its claim of copyright infringement, and held that Atari's own "unclean" hands barred any defense of copyright misuse.¹⁰⁰ However, the court of appeals held that Atari's initial, unsuccessful reverse engineering efforts, untainted by the purloined copy and necessary to understanding the Nintendo program, constituted a fair use of Nintendo's lockout program.¹⁰¹

In contrast to Atari's reverse engineering efforts, Accolade's first round of reverse engineering did not focus on the lockout program that Sega ultimately installed in its "Genesis III" console. Instead, Accolade wired a decompiler into an original "Genesis" console to retrieve object code and generate printouts of source code for the purpose of discovering the interface specifications for the console.¹⁰² This decompilation enabled Accolade to create and adapt its own games for use on that console.¹⁰³

Meanwhile, Sega developed a new "Genesis III" console, which included a lockout device.¹⁰⁴ Accolade learned of the impending release of the "Genesis III" console in the United States and discovered at a consumer electronics show prior to the release of that console that its "Genesis"-compatible game cartridges would not operate on "Genesis III."¹⁰⁵ Accolade then initiated a second round of reverse engineering, this time focusing on a small segment of code of unidentifiable function found in the "power-up" sequence of every Sega game.¹⁰⁶ Accolade added the segment as a standard header file for all of its "Genesis"-compatible games, and as Accolade had suspected, the segment provided the "key" that unlocked the "Genesis III" console.¹⁰⁷ Unbeknownst to Accolade, however, the segment also prompted a visual display that indicated that the inserted cartridge was produced by or under license from Sega.¹⁰⁸

Accordingly, Sega filed suit against Accolade, claiming trademark infringement and false designation of origin, in violation of §§ 32

(1) and 43(a) of the Lanham Act.¹⁰⁹ Sega later amended its complaint to include a claim of copyright infringement for Accolade's successful reverse engineering of the lockout device for Sega's game cartridge programs.¹¹⁰ Although the district court rejected Accolade's functionality defense to Sega's claim of trademark infringement and its fair use defense to Sega's copyright infringement claim,¹¹¹ the court of appeals reversed and held for Accolade on both counts.¹¹²

In contrast to the lockout programs that were reverse engineered in *Atari* and *Sega*, the program reverse engineered in *Vault* was a copy-protection program that Vault included on otherwise blank "PROLOK" computer diskettes.¹¹³ The copy-protection program was designed to prevent purchasers of programs recorded on a "PROLOK" diskette from making unauthorized copies of the recorded program for distribution to others.¹¹⁴ Quaid analyzed the operation of the "PROLOK" diskette using "Disk Explorer" and "IBM Debug," two products sold on the open market.¹¹⁵ Although Quaid did at one point disassemble the "PROLOK" program, the product developed by this means was subsequently discontinued. Quaid then developed a program which allowed it to analyze the functions of various programs without disassembling or decompiling them.¹¹⁶ Using this program, Quaid developed a diskette called "CopyWrite," containing a feature called "RAMKEY," which could unlock the "PROLOK" protection device and facilitate the creation of a fully functional copy of a program recorded on a "PROLOK diskette." Both the district court and Court of Appeals rejected Vault's claim that Quaid's program infringed Vault's copyright on its copy protection program and contributed to Quaid's customers' infringement of copyright on the programs of Vault's customers.¹¹⁷

B. A Comparison of *Atari*, *Sega*, and *Vault* with the European Community Directive

The court of appeals decisions in *Atari*, *Sega* and *Vault* appear to have had the collective effect of harmonizing United States copyright law with the recent European Community Directive on the Legal Protection of Computer Programs.¹¹⁸ Article 5(3) of the EC Directive specifies that:

The person having a right to use a copy of a computer program shall be entitled, without the authorization of the [copy] rightholder, to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he is entitled to do.¹¹⁹

This provision, by allowing the reverse analysis of a computer program while using it in conjunction with a computer, appears to create a right equivalent to the one recognized in *Vault* as emanating from § 117 of the U.S. Copyright Act, to engage in reverse analysis "as an essential step in the utilization of a program."

In addition to Article 5(3), the EC Directive contains a specific provision governing the permissibility of decompilation or disassembly. Article 6 of the EC Directive specifies that:

The authorization of the [copy]rightholder shall not be required where reproduction of the code and translation of its form . . . are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided that the following conditions have been met:

- (a) these acts are performed by the licensee or by another person having a right to use a copy of a program, or on their behalf by a person authorized to do so;
- (b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in subparagraph (a);
- (c) these acts are confined to the parts of the original program which are necessary to achieve interoperability.¹²⁰

Article 6 thus creates a right to decompile or disassemble a computer program where it is necessary to achieve interoperability with another program. This provision appears to create a right equivalent to the one recognized in *Atari* and *Sega* as fair use under § 107, to reverse engineer a computer program in order to defeat a lockout device.

The following subsections will examine the precise scope of the reverse engineering privilege under §§ 107 and 117 as spelled out in the *Atari*, *Sega* and *Vault* cases, and will compare these cases with the reverse engineering provisions contained in the EC Directive.

C. Reverse Engineering as a Fair Use under § 107

The first provision of the 1976 Copyright Act applicable to reverse engineering of computer programs is § 107, which provides that "the fair use of a copyrighted work, including such use by reproduction in copies . . . for purposes such as . . . teaching . . . scholarship or research, is not an infringement of copyright."¹²¹ Section 107 goes on to enumerate four non-exclusive factors that courts may consider in determining whether a particular use is a fair use. These factors are:

- (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 amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
- (4) the effect of the use upon the potential market for or value of the copyrighted work.¹²²

*Atari*¹²³ and *Sega*¹²⁴ were the first court of appeals decisions to apply these four factors in a case specifically involving reverse engineering of computer programs. At least two prior district court decisions, however, seem to have assumed that reverse engineering is or might be a fair use of a copyrighted program, just as the district courts in *Atari* and *Sega* assumed or specifically held that reverse engineering of a computer program was not a fair use.

1. Prior District Court Case Law

In *E.F. Johnson Co. v. Uniden Corp. of America*,¹²⁵ the court's finding of infringement was based on the verbatim reproduction of substantial sections of the plaintiff's code, as well as other indications of substantial similarity between plaintiff's and defendant's programs.¹²⁶ The court, however, stated by way of footnote dictum that:

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 competitors' 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. . . While defendant may have permissibly dumped, flow charted and analyzed plaintiff's code, it could not permissibly copy it.¹²⁷

Conversely, in *NEC Corp. v. Intel Corp.*,¹²⁸ the court found no infringement where the creator of two of NEC's microcodes testified that although he had disassembled two of Intel's microcodes, he did not undertake to copy them.¹²⁹ The court stated that, while the creator of NEC's microcodes might have made reference to the Intel microcodes, this activity did not result in "slavish copying."¹³⁰

The district court in *Atari*, on the other hand, had held that Atari was not free to appropriate Nintendo's specific codes used for "locking" the Nintendo game console to prevent all but Nintendo-authorized game cartridges from being played on the console.¹³¹ In so holding, the court acknowledged the footnote dictum in *E.F. Johnson Co.*, but read it as giving approval only to "surveying the general outline of an existing program."¹³² The court went on to hold that intermediate copying (i.e., decompilation or disassembly) of a program in the course of reverse engineering would constitute direct infringement, and that Atari's final program incorporating Nintendo's codes to enable Atari game cartridges to be played on the Nintendo game console was an infringing derivative work (i.e., adaptation).¹³³

Likewise, the district court in *Sega* held that if the process of reverse engineering entails the duplication of the copyrighted work and the recasting or transformation of the object code into a form more intelligible to humans, it may infringe upon the copyright owner's exclusive rights.¹³⁴ The district court found that Accolade infringed Sega's copyright when it reverse engineered Sega's game cartridges and copied code that was required to unlock the "Genesis III" console.¹³⁵ The court distinguished *NEC Corp. v. Intel Corp.* and *E.F. Johnson Co. v. Uniden Corp. of America*, by pointing out that neither case specifically addressed the issue of intermediate copying, and that the footnote dictum in the latter case stated only that dumping and analyzing of competitors' codes would not, in itself, establish piracy.¹³⁶ The court further noted that if Congress had intended to create an exception for intermediate copying, it would have provided for it as it did in the Semiconductor Chip Protection Act.¹³⁷

2. The Relevance of the Semiconductor Chip Protection Act

Four years after the Computer Software Copyright Act of 1980¹³⁸ amended the 1976 Copyright Act to make computer programs explicitly copyrightable,¹³⁹ Congress created a *sui generis* body of federal intellectual property law to protect semiconductor chip designs.¹⁴⁰ Although the Semiconductor Chip Protection Act of 1984¹⁴¹ makes it an infringement to reproduce a chip design or to import or distribute a semiconductor chip product embodying the design, § 906(a) of the Act expressly states that:

it is not an infringement of the exclusive rights of the owner of a mask work for-

(1) a person to reproduce the mask work solely for the purpose of teaching, analyzing, or evaluating the concepts or techniques embodied in the mask work or the circuitry, logic flow, or organization of components used in the mask work; or

(2) a person who performs the analysis or evaluation described in paragraph (1) to incorporate the results of such conduct in an original mask work which is made to be distributed.¹⁴²

The legislative history of the Chip Act states that it is an established industry custom in the United States to photograph chips produced by competitors in order to design similar, hopefully improved, chips.¹⁴³ The custom is deemed fair as long as "substantial analysis and study," rather than simple plagiarism, occurs, because it produces multiple sources for a single chip and promotes competition in the industry as a whole.¹⁴⁴

It could be argued that because § 906(a) of the Chip Act, explicitly permits reproduction of semiconductor chip designs for analysis, § 107 of the Copyright Act may be interpreted as implicitly permitting the intermediate copying of computer programs by analogy. The problem with this argument is that the opposite conclusion is equally plausible, and was in fact precisely the argument adopted by the district court in the *Sega* case. The *Sega* court stated that, had Congress intended to create a fair use exception for intermediate copying of computer programs, "it would have provided for it as it did in the Semiconductor Chip Protection Act. . . ."¹⁴⁵

The legislative history of the Chip Act further reveals that Congress initially considered amending the 1976 Copyright Act to provide copyright protection for semiconductor chip designs. The legislative history accompanying the earliest of the chip-design protection proposals contained conflicting views on whether reverse engineering of a chip design would fall within the § 107 fair use privilege.¹⁴⁶ Some legislators were concerned that defining reverse engineering of a chip design as fair use might encourage a more expansive interpretation of fair use as applied to literary works, such as computer programs.¹⁴⁷ Thus, Congress considered creating an express reverse-engineering right, independent of § 107, for semiconductor chip designs.¹⁴⁸

Eventually, Congress altogether abandoned the effort to extend copyright protection to semiconductor chip designs-not because of any concern that to grant such protection might lead to an expansive interpretation of § 107's fair use privilege to permit reverse engineering of computer programs, as the district court in *Sega* erroneously suggested,¹⁴⁹ but, because it believed that semiconductor chips were intrinsically utilitarian articles that were beyond the scope of the Copyright Act.¹⁵⁰ Rather, Congress opted for a *sui generis* form of protection that contained an express reverse engineering privilege identical to the one originally proposed as an amendment to the 1976 Copyright Act. The Chip Act accordingly omitted any mention of an exclusive right to prepare adaptations of semiconductor chip designs,¹⁵¹ and disclaimed any intention to affect the scope of copyright protection for computer programs.¹⁵²

By contrast, the 1980 amendment to the Copyright Act included neither an analogous reverse-engineering provision nor eliminated the copyright holder's exclusive right to make adaptations of computer programs. Rather, Congress amended § 117 of the 1976 Copyright Act to include an explicit, limited right of owners of copies of copyrighted computer programs to reproduce or adapt the programs (1) as an essential step in the utilization of a computer program in conjunction with a machine, and (2) in order to make archival copies of the program.¹⁵³

These differences between the Semiconductor Chip Act and the amended 1976 Copyright Act suggest that courts should not construe § 107 to include a blanket reverse-engineering privilege. On the other hand, just because the 1976 Copyright Act contains no explicit reverse engineering privilege, and § 117 of the Act explicitly permits reproduction or adaptation of a copyrighted computer program in only two limited circumstances, it does not follow that all other unauthorized uses of a computer program will necessarily constitute an infringement. Both the language and the legislative history of the 1976 Copyright Act indicate that some uses other those set forth in §§ 108-120¹⁵⁴ may nevertheless fall within § 107's more general fair use privilege. Moreover, as the court of appeals in *Atari* pointed out, the legislative history of § 107 specifically states that courts should interpret the fair use exception to accommodate new technological innovations.¹⁵⁵

In summary, the most that can be said of the differences between the 1976 Copyright Act, as amended in 1980, and the Semiconductor

Chip Act of 1984 is that Congress intended to do exactly what it did in these two Acts. Congress explicitly permitted the reverse engineering of semiconductor chip designs, while subjecting reverse engineering of computer programs to the four-factor test set out in § 107, or, in the alternative, to the requirements set out in § 117 for the two limited uses explicitly allowed by that section.

3. *The Four-Factor Test: The Commentators and the Atari and Sega Decisions*

Atari and *Sega* were the first cases that explicitly applied § 107's four-factor test to allow particular instances of reverse engineering. However, a number of commentators, as well as the federal district court in *Sega* itself, had previously applied § 107's four factors to determine the legality of reverse engineering and arrived at a different conclusion. As early as 1985, William Patry, the commentator principally cited by the *Sega* district court, concluded that:

A quick examination of typical reverse engineering under the statutory four fair use factors . . . indicates that fair use is not available. The nature and purpose of the use is entirely commercial; the copyrighted source code is an unpublished work subject to a narrow scope of fair use; the entire work is copied; and since the use is commercial, potential harm to the market for the original is presumed.¹⁵⁶

More recently, the Committee on Computer Law, organized by the Association of the Bar of the City of New York, issued a report entitled "Reverse Engineering and Intellectual Property Law"¹⁵⁷ (hereinafter the "Committee Report"). In the report, the Committee more thoroughly analyzed examples of reverse engineering to determine whether they were fair use within the meaning of § 107. The Committee concluded that "it is difficult to say categorically whether copies created through decompilation in any of our examples are 'fair use'. . ."¹⁵⁸ Instead, the Committee tried to indicate what considerations should be weighed by a court making such a determination.

In the same year, a second report emerged as a result of the "LaST Frontier Conference On Copyright Protection of Computer Software."¹⁵⁹ This report concluded that

making a small number of copies for the sole purpose of studying a copyrighted program for possible use of its unprotected elements, should be considered a necessary step in allowable examination or study and should, therefore, be deemed not to infringe the copyright under the fair use provisions of the Copyright Act."¹⁶⁰

Specifically, the Conference Report concluded: 1) that "although the eventual use is commercial, the copying is effected for the socially desirable research and development of a competing product," 2) that the copyrighted work is in a "form that cannot be examined except through a taking that would otherwise constitute infringement," 3) that although the entire work is copied, the Supreme Court has held that using a video recorder to copy an entire copyrighted television broadcast can be a fair use; and 4) that the limited use in question does not itself have any material effect on either the market for or value of the copyrighted work and that even though a non-infringing competing work may result from the reverse engineering and adversely affect the value of the original work, it is not then the copy that is competing.¹⁶¹

Shortly after the *Atari* and *Sega* cases were decided, Professor Arthur Miller, a distinguished professor of intellectual property law at Harvard University, and a former member of the commission which recommended the amendments to the Copyright Act making it explicitly applicable to computer programs, published an article highly critical of the decisions in *Atari* and *Sega*.¹⁶² In this article, Professor Miller essentially reiterates the arguments of earlier courts and commentators who would hold that decompilation and disassembly do not constitute a fair use of copyrighted works.

The views of the foregoing commentators represent but one facet of a larger ongoing-and occasionally and vituperative¹⁶³-second generation (see *supra* Part III(C)(2)) debate over the appropriate scope of copyright protection for computer programs. Aside from the court of appeals decisions in *Atari* and *Sega*, the Committee Report provides the most extensive and balanced discussion of § 107's four-factor test for whether reverse engineering is fair use. However, the Committee Report comes to conclusions that differ significantly from the opinions in *Atari* and *Sega*, as well as the EC Directive. The following subsections compare the Committee Report, as well as the views of other courts and commentators, with the *Atari* and *Sega* decisions and the EC Directive in order to illuminate the full range of factors that should be considered by courts in applying § 107's four-factor test to reverse engineering of computer programs.

a) The First Factor: The Purpose and Character of the Use

The language of § 107(1) indicates that in determining the purpose and character of the use of the copyrighted work, courts should

specifically consider whether the use is of a commercial nature or is for nonprofit, educational purposes.¹⁶⁴ The district court in *Sega* concluded that Accolade's reverse engineering was presumptively unlawful because it was undertaken for financial gain and was aimed at the creation of a competitive product that would adversely affect the value of the copyrighted work.¹⁶⁵

Similarly, the Committee Report pointed out that commercial use is explicitly mentioned in § 107.¹⁶⁶ The Committee Report stated that "[e]very commercial use of copyrighted material is presumptively . . . unfair."¹⁶⁷ It noted that § 107 lists "teaching, scholarship, and research" as examples of activities in which a finding of fair use is most appropriate.¹⁶⁸ The Committee concluded that the use of decompilation to develop a competing program, or to facilitate the making of infringing copies of the original program, would weigh against a finding of fair use.¹⁶⁹ On the other hand, the Committee pointed out that the fair use doctrine nevertheless has been applied in commercial contexts, and that commercial use does not preclude a finding of fair use.¹⁷⁰

The Committee's examination of the first factor, however, fails to discuss the immediate purpose served by decompilation or disassembly, namely, to develop a limited number of human-readable copies of the program in order to study its structure, sequence, and organization. Whether the ultimate purpose is commercial or non-commercial, the immediate purpose of reverse engineering is to gain access to unprotected ideas.¹⁷¹

The district court in *Sega* rejected this characterization of the purpose of reverse engineering, stating that it "misconceives the role of both Section 102(b) and Section 107 of the Copyright Act and does violence to the term 'access.'"¹⁷² The court stated that "the emphasis on appropriating the idea mistakes the role of § 102(b), which denies protection *ab initio* to ideas but which is not designed to form an independent basis for the appropriation of protectible expression under the cloak of § 107."¹⁷³

This criticism misstates the issue. The fact that § 102(b) may not have been designed to form an independent basis for the appropriation of expression under § 107 does not undercut the argument that the four factor test of § 107 itself provides an adequate basis for determining the fairness of appropriating copyrighted expression in order to gain access to the unprotected ideas that remain inaccessible even after a program is published.

Nor does it refute the user's claimed right of access to a work's ideas or unprotectible expression to say, as the district court in *Sega* does, that the "public's need for access to the copyrighted work is fully satisfied by the copyright owner's marketing of the original." Access to a copyrighted computer program does not necessarily provide access to the unprotectible ideas that it contains, unless one assumes that the only unprotected ideas in a computer program are those functions that are apparent in the normal use of the program. This conception of a computer's unprotected ideas is essentially that adopted in *Whelan Associates v. Jaslow Dental Laboratory*, the first court of appeals decision in the second generation of computer copyright cases.¹⁷⁴ However a second court of appeals decision, *Computer Associates International v. Altai, Inc.*, rejects *Whelan's* formulation as descriptively inadequate.¹⁷⁵ As noted by the Committee Report in its discussion of the second fair use factor, prohibiting decompilation could give the copyright owner "a monopoly over the ideas, as well as the expression in the copyrighted work,"¹⁷⁶ thus defeating the policy objective of improving and refining ideas contained in computer programs.

Also unconvincing is the *Sega* district court's assertion that a competitor who reverse engineers a copyrighted computer program is "not at all interested in increasing that access. . . his only purpose is to get the public to purchase his work rather than the original."¹⁷⁷ This assertion simply fails to recognize the reverse engineer's right, as a member of the public, to have access to the work's ideas. Nor is it true as the *Atari* and *Sega* cases illustrate, that the purpose of a competitor who reverse engineers a copyrighted computer program is merely to get the public to purchase his work rather than the original. In both *Atari* and *Sega*, the defendant's purpose was to make its game cartridges compatible with its competitor's consoles. As the court of appeals in *Sega* took pains to point out, there was no basis in the record for assuming that Accolade's game cartridges would significantly affect the market for Sega's game cartridges, since a consumer "might easily purchase both."¹⁷⁸ Even if the market were affected, however, the fairness of Accolade's use of the product of its reverse engineering should turn on a separate four-factor inquiry into the fairness of that use.¹⁷⁹

The courts of appeals in both *Atari* and *Sega* recognized that the immediate purpose motivating reverse engineering of computer programs should be at least as important as its ultimate, commercial purpose. The court in *Atari* held that § 107 of the Copyright Act "permits an individual in rightful possession of a copy of a work to undertake necessary efforts to understand the work's ideas, processes, and methods of operation." The court also noted that the reason § 107 exempts from copyright protection reproductions for criticism, comment, or research is because "these activities permit public understanding and dissemination of the ideas, processes, and methods of operation in a work."¹⁸⁰

Likewise, the court in *Sega* stated that the use at issue "was an intermediate one only and thus any commercial 'exploitation' was indirect and derivative."¹⁸¹ The court went on to note that:

although Accolade's ultimate purpose was the release of Genesis-compatible games for sale, its direct purpose in copying Sega's code, and thus its direct use of the copyrighted material, was simply to study the functional requirements for Genesis compatibility so that it could modify existing games and make them usable with the Genesis console.¹⁸²

The court in *Sega* also stated that Accolade's reverse engineering "led to an increase in the number of independently designed video game programs offered for use with the Genesis console."¹⁸³ The court concluded that it is "precisely this growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote."¹⁸⁴ The foregoing observation emphasizes that, irrespective of the reverse engineer's "purpose," a court's determination of the "character" of the use under the first fair use factor should turn on whether the reverse engineering is productive in that it leads to the creation of additional works of authorship.

(1) The Relevance of the Copyright Misuse Defenses

Although offering a fuller discussion of the first fair use factor than the commentators did, the courts of appeals in *Atari* and *Sega* nevertheless failed to identify explicitly one key aspect of the purpose motivating the reverse engineering involved in those two cases. Both Atari and Accolade were attempting to defeat lockout devices by Nintendo and Sega, the employment of which may have constituted copyright misuse. Ironically although Atari and Accolade unsuccessfully raised other claims of copyright misuse on the part of Nintendo and Sega, neither appears to have claimed that the lockout device constituted copyright misuse.¹⁸⁵

In considering the misuse claims raised in *Atari*, the court of appeals recognized that several other courts of appeals had previously entertained defenses of copyright misuse, analogizing from the well-established patent misuse defense. The court noted that although only one decision had actually sustained the defense, the United States Supreme Court had previously given at least tacit approval of it.¹⁸⁶ However, the court concluded that, in the absence of statutory entitlement, the copyright misuse defense is only an equitable doctrine that is unavailable to a party that does not have "clean hands."¹⁸⁷ Because Atari had misled the Copyright Office in order to obtain a copy of Nintendo's registered program, it was ineligible to invoke the defense.¹⁸⁸

In *Sega*, the district court said only that Accolade's misuse defense was based upon antitrust tying allegations, which the court ordered stricken from Accolade's counterclaim.¹⁸⁹ The court concluded that "[n]o antitrust violation is alleged, nor is there proof of fraud or other clear violation of a legal duty. . . . Thus, Accolade has failed to establish a copyright misuse defense."¹⁹⁰

Although the court of appeals in *Sega* made no explicit reference to copyright misuse, it nevertheless seems to have implicitly recognized (but not in its discussion of the first fair use factor) that Accolade had a plausible copyright misuse defense.¹⁹¹ Had Atari's own hands been "cleaner," it appears that it too would have had such a defense.

By creating lockout devices, Nintendo and Sega were arguably attempting to maintain a monopoly on the market for video game cartridges compatible with their respective game consoles.¹⁹² Because game cartridges compatible with the console of one of these two giants in the field of video entertainment systems were not compatible with the console produced by the other, Nintendo and Sega each had the power by virtue of their ability to lock out competitors' game cartridges, to raise the price of cartridges, provided only that prices not reach a level that would induce owners of Nintendo or Sega "Genesis" consoles to purchase the competing console.¹⁹³ By concealing the codes that unlock their consoles and invoking copyright protection for these codes, Nintendo and Sega unequivocally engaged in practices aimed at barring the entry of competitors into their respective cartridge markets.

Significantly, while discussing the fourth fair use factor, the court of appeals in *Sega* remarked that

an attempt to monopolize the market by making it impossible for others to compete runs 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.¹⁹⁴

This remark offers support for the conclusion that, not only was the purpose and character of Accolade's use a fair one, but Sega's use of the lockout device constituted a misuse of its copyright.

(2) The Relevance of the Trademark Misuse Defense

Although the court of appeals did not explicitly recognize a copyright misuse defense, it did hold that Sega designed its computer program in such a way as to misuse its trademark.¹⁹⁵ The same security system that locked the Sega Genesis console also triggered a

visual display, once the console was unlocked, which read "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD."¹⁹⁶ The codes that Accolade reverse engineered from Sega's game cartridges and included in its own game cartridges caused this misleading message to be displayed when Accolade's game cartridges were inserted into Genesis III consoles.¹⁹⁷ Creating such a risk of confusion, the court said, might be sufficient to constitute trademark misuse. Additionally, the court held that a trademark is misused if it serves to limit competition in the manufacture and sales of a product.¹⁹⁸

In sum, the foregoing discussion suggests that a particularly compelling reason for finding reverse engineering to be a fair use exists when the program is itself being misused by its developer.

b) The Second Factor : The Nature of the Copyrighted Work

In examining the second factor, the Committee Report belatedly acknowledged that,

unlike a book, whose ideas can be discovered by reading it, the nature of a computer program distributed only in object code is such that the ideas embodied in it largely cannot be studied and understood by humans without decompilation.¹⁹⁹

As we have seen, the Committee notes that prohibiting decompilation could defeat an important societal interest in learning from and improving on ideas contained in computer programs, "thereby giving the copyright owner a monopoly over the ideas, as well as the expression in the copyrighted work."²⁰⁰ Thus, the Committee concludes that this factor generally weighs strongly in favor of a finding of fair use. However, if this is true, it would appear that the first factor should also weigh more strongly in favor of a finding of fair use than the Committee recognizes,²⁰¹ since the immediate purpose served by decompiling or disassembling a computer program is precisely to gain access to, learn from, and perhaps improve on the ideas contained in the computer program.

The courts of appeals in *Atari* and *Sega* also recognized that a computer program's "nature" is unique. As the court in *Sega* pointed out, although the "unprotected aspects of most functional works are readily accessible to the human eye . . . humans often cannot gain access to the unprotected ideas and functional concepts contained in object code without disassembling that code-i.e. making copies."²⁰² Consequently, the court of appeals in *Atari* reasoned that "when the nature of a work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying."²⁰³

What neither the *Atari* court nor the *Sega* court pointed out is that the Copyright Office itself contributes to what these courts identify as the unique nature of computer programs. First, as the facts of the *Atari* case indicate, Copyright Office regulations limit the circumstances in which works deposited in connection with registration can be reproduced.²⁰⁴ In addition, certain other Copyright Office regulations, clearly authorized by the 1976 Act but adopted by the Copyright Office with little apparent consideration of the underlying policy consequences, permit those registering computer programs with the Copyright Office to deposit only identifying portions of the source code version of the program, or even only the object code.²⁰⁵ These Copyright Office regulations in effect create a federal form of trade secret protection for computer programs, but seem inconsistent with the underlying policies of the Copyright Act, and thus may be in excess of the Copyright Office's statutory authority.²⁰⁶ Ironically, the promulgation of these regulations strengthens the argument of users that no alternative means exist for obtaining access to a computer program's unprotected ideas.

An additional aspect of the nature of computer programs, not mentioned by the Committee Report, but buttressing its general conclusion that this factor weighs in favor of a finding of fair use, is that computer programs may be regarded, wholly or in part, as utilitarian or factual works. Long before the fair use exception was codified in § 107, the Supreme Court in *Baker v. Selden*²⁰⁷ suggested that works having strong functional elements, such as accounting textbooks, were not entitled to the same degree of protection as non-factual works. The Court also suggested that some particularly utilitarian elements of a work, such as blank accounting forms, may not be entitled to copyright protection at all.²⁰⁸

More recently, in *Harper & Row, Publishers v. Nation Enterprises*,²⁰⁹ the Supreme Court explicitly recognized, in applying the second factor of § 107's fair use test to an unpublished²¹⁰ historical narrative or autobiography, that copyright law "generally recognizes a greater need to disseminate factual works than works of fiction or fantasy."²¹¹ Likewise, in *Feist Publications v. Rural Telephone Service Co.*, the Court noted that works that are merely compilations of fact are copyrightable, but copyright protection for such a work is "thin."²¹²

The court of appeals in *Sega* recognized that computer programs pose unique problems because while there may be many ways for a computer programmer to accomplish a given task or fulfill a particular market demand, making the programmer's choice of program structure and design highly creative and idiosyncratic, computer programs are nevertheless essentially utilitarian articles that

accomplish particular tasks.²¹³ Computer programs thus contain many logical, structural, and visual display elements that are dictated by function, efficiency, and external factors, such as compatibility requirements and industry demands.²¹⁴

One further aspect of a computer program's particular nature must be noted. The Supreme Court in *Harper & Row* recognized that whether a work is published or unpublished is also a relevant aspect of its nature, and that the scope of fair use is narrower with respect to unpublished works. In that very statement, of course, the Court seems to have conceded that the fair use doctrine applies to unpublished works.²¹⁵ But nothing in *Harper & Row* indicated how the Court would deal with a work such as a computer program, which can be both published (in its object-code form) and unpublished (in its source-code form). The court in *Sega*, on the other hand, specifically noted that in one of its earlier decisions, *Lewis Galoob Toys v. Nintendo of America*, it had affirmed a district court holding that computer game cartridges that are held out to the public for sale are published works for purposes of copyright.²¹⁶

c) The Third Factor : The Amount and Substantiality of the Portion of the Copyrighted Work Used

Regarding the third factor, courts and commentators agree that the results of decompilation are likely to be both quantitatively and qualitatively "substantial" in relation to the copyrighted work as a whole.²¹⁷ This is because disassembly or decompilation usually entails making a nearly complete copy or adaptation of a program.²¹⁸ However, the Committee Report states that even complete verbatim copies may be non-infringing.²¹⁹ The Report noted that in *Sony Corporation v. Universal Studios*, the Supreme Court held that the videotaping of entire copyrighted television broadcasts for later home viewing was a fair use.²²⁰ In that case, the purpose of the use was more determinative of its fairness than the amount of copyrighted expression used. Even though the use of the copyrighted work in *Sony* was non-productive, it was also for a non-commercial purpose, which weighed heavily in favor of a finding of fair use.²²¹

Where one who reverse engineers a computer program uses portions of protected expression from the decompiled work in the production of a competing work, the fairness of that use should turn on a separate determination of the amount and substantiality of the protected expression used in relation to the copyrighted work as a whole. Admittedly, as the court of appeals in *Atari* stated, "[f]air use to discern a work's ideas . . . does not justify extensive efforts to profit from replicating protected expression."²²² As the facts of the *Atari* and *Sega* cases illustrate, however, the amount of protected expression ultimately used, when the purpose of the reverse engineering is simply to discover codes that will unlock a game console, is likely to be minimal.²²³ Even if the code used may be considered a qualitatively substantial, albeit quantitatively small part of the decompiled program, it is substantial precisely because of the utilitarian function that it plays and is thus arguably unprotectible expression.

d) The Fourth Factor: The Effect on the Potential Market for or Value of the Copyrighted Work

In considering the fourth factor, the Committee Report cites the Supreme Court's opinion in *Harper & Row* for the proposition that market impact is "the single most important element of fair use."²²⁴ The Committee concludes without further analysis that this factor would not support a claim of fair use if the "effect of the use is the creation of a competing program."²²⁵ The Committee merely cites the legislative history of the Copyright Act for the proposition that "a use which supplants any part of the normal market for a copyrighted work would ordinarily be considered an infringement."²²⁶ Unfortunately, the Committee's examination of this final, and most important factor is the most superficial part of its fair use analysis. The Committee fails to acknowledge that the copyrighted work exists in two different forms, each with its own particular potential market or value.

Decompiling the object-code version of a computer program generally has no direct effect on the potential market for the program in object-code form, because the decompiled version of the program cannot be used to operate a computer.²²⁷ The only direct market effect of decompilation is on the potential market for the source-code version of the program. Yet, much of the value of the source-code version of the program comes, not from the manner in which it is expressed, but from the fact that the source code, and the uncopyrightable ideas or functional expression that it contains, can be kept secret, even after the object-code version of the program is publicly distributed.²²⁸

The fairness of reverse engineering should not be confused with the fairness of the ultimate uses made of the product of the reverse engineering or with the potential effects of such uses on the market for the publicly distributed program. And yet, the Committee Report seems to make precisely this mistake when it concludes that the fourth factor of § 107's fair use test would not support a finding of fair use where "the effect of the use is the creation of a competing program."²²⁹ This view fails to distinguish between creating a competing program that uses copyrighted expression contained in the copyrighted program, and creating a competing program that merely accomplishes the same function or uses unprotected ideas found in the copyrighted program. While the former situation would raise a legitimate fair use issue, the latter would not. Copyright protection extends only to an author's expression and not to the ideas expressed. The fair user "may make significant use of prior work, so long as he does not bodily appropriate the expression of another."²³⁰

It is at this point that the second generation of copyright cases concerned with the scope of protection available for computer programs becomes critical to the question of the permissibility of reverse engineering. For the broader the scope of protection for non-literal elements of a program, the narrower the universe of ideas available to others. Conversely, if the scope of protection is narrow, the universe of unprotected ideas is much broader.

It is in this connection that the court of appeals in *Sega* cites with approval the recent decision of the Court of Appeals for the Second Circuit in *Computer Associates Int'l, v. Altai, Inc.*²³¹ *Computer Associates* held that a computer's ultimate function or purpose is the composite result of interacting subroutines, each of which may represent its own idea, and may contain purely utilitarian elements that can only be expressed in a limited number of ways. As we have seen, the court in *Computer Associates* found that the formulation in *Whelan Associates v. Jaslow Dental Laboratory*²³² which holds that the idea or function of a computer program is the idea of the program as a whole, was descriptively inadequate.²³³ If *Altai*, rather than *Whelan*, comes to represent the culmination of the second generation of computer program copyright cases, the universe of unprotected ideas, to which reverse engineering could provide access, will be substantial.

4. *Atari, Accolade and The European Community Directive*

In *Atari* and *Sega*, there were particularly compelling reasons for finding that reverse engineering designed to defeat a lockout device constitutes a fair use of a computer program, namely, to defeat another's misuse of its copyrighted work.²³⁴ Reverse engineering to defeat a lockout device, however, is itself but a particular species of the more general effort to achieve interoperability—precisely the purpose of which Article 6 of the European Community Directive speaks in recognizing a specific right to decompile or disassemble another's copyrighted computer program. Thus, analysis of the fairness of other examples of reverse engineering for the purpose of achieving interoperability should yield essentially the same results as were reached in *Atari* and *Sega*, aside from the copyright misuse aspect of those cases. As Article 6 of the EC Directive makes clear, the principal focus of such an analysis should be on whether reverse engineering is "indispensable to obtain the information necessary to achieve the interoperability of an independently created program." If this requirement is met, decompilation or disassembly will be fair use under § 107 of the Copyright Act.

It will be noted that Article 6 of the EC Directive specifically permits decompilation or disassembly "to obtain the information necessary to achieve the interoperability of an independently created computer program *with other programs*."²³⁵ The legislative history of the EC Directive makes clear that the foregoing italicized phrase, which has been described as having "engendered more controversy than anything else in the EC Directive," was settled upon in preference to alternative language which would have permitted decompilation solely to achieve interoperability with the "decompiled" or "original" program.²³⁶ The purpose of the language change was to make it clear that Article 6 permitted decompilation for the purpose of developing a competing work.²³⁷

Atari and *Sega* do not explicitly address whether § 107 of the Copyright Act would permit reverse engineering for the ultimate purpose of creating a competitive product.²³⁸ However, the language of the EC Directive, the reasoning in *Atari* and *Sega*, and the foregoing critique of the Committee Report all suggest that the § 107 privilege to reverse engineer computer programs may not be limited to instances in which the purpose of the reverse engineering is to achieve interoperability with the program being reverse engineered. The fairness of reverse engineering for the purpose of developing a competing program, of course, should turn not only on whether the reverse engineering itself is indispensable, but on whether and how much copyrightable expression is used in the competing program.

On the other hand, future cases may construe *Atari* and *Sega* as having held only that reverse engineering is fair when it is attempting to defeat the kind of copyright misuse that lockout devices represent.²³⁹ Given the continuing uncertainty as to the exact parameters of the fair-use privilege to reverse engineer computer programs in the United States, it is appropriate to examine § 117, the other relevant provision of the 1976 Copyright Act that has been applied to permit reverse engineering of computer programs.

D. Section 117: Limitations on Exclusive Rights in Computer Programs

Section 117 of the amended 1976 Copyright Act states that:

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

- (1) that such new copy or adaption 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 adaption 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.²⁴⁰

Congress enacted § 117 almost exactly in the form recommended by the National Commission on New Technological Uses of Copyrighted Works (CONTU).²⁴¹ The only modification was that the holder of the right was changed from a "rightful possessor" to the "owner" of a copy of the computer program.²⁴² According to the CONTU report, a "rightful possessor" was to include not only a purchaser but also a lessee or licensee of a copy of a computer program.²⁴³ Because the 1980 Act's legislative history merely states that the Act "embodies the recommendations of [CONTU] with respect to clarifying the law of copyright of computer software,"²⁴⁴ the courts have tended to treat the Final Report of CONTU as legislative history.²⁴⁵

1 The CONTU Report and Congressional Modification of CONTU's Proposed § 117

In its report, CONTU stated that, "[b]ecause the placement of a work into a computer is the preparation of a copy, the law should provide that persons in rightful possession of copies of programs be able to use them freely without fear of exposure to copyright liability."²⁴⁶ Although in the vast majority of cases, "creators, lessors, licensors, and vendors of copies of programs intend that they be used by their customers," CONTU anticipated situations "in which the copyright owner might desire, for good reason, or none at all, to force a lawful owner or possessor of a copy to stop using a particular program."²⁴⁷ To prevent this from happening, CONTU designed § 117 to provide a rightful possessor of a program "with a legal right to copy it to that extent which will permit its use by that possessor."²⁴⁸ Additionally, CONTU thought that a rightful possessor of a copy of a program should have the right "to load it into a computer and to prepare archival copies of it to guard against destruction or damage by mechanical or electrical failure."²⁴⁹

CONTU also proposed a limited right of adaptation, recognizing that

[b]ecause of a lack of complete standardization among programming languages and hardware in the computer industry, one who rightfully acquires a copy of a program frequently cannot use it without adapting it to that limited extent which will allow its use in the possessor's computer.²⁵⁰

Thus, § 117 was intended to provide "a right to make those changes necessary to enable the use for which [the program] was both sold and purchased."²⁵¹ CONTU gave two examples of permissible adaptation: "The conversion of a program from one higher level language to another to facilitate use," and "the right to add features to the program that were not present at the time of rightful acquisition."²⁵²

To protect the interests of the copyright owner, CONTU specified that exact copies made pursuant to § 117 were to be leased, sold or otherwise transferred only in connection with the transfer "of all rights in the program, thus creating a new rightful possessor and destroying that status as regards the seller."²⁵³ Adaptations, by contrast, were purely for the personal use of party making the adaptation and could not be transferred without the permission of the copyright owner. Analogizing such adaptations to marginal note-taking in a book, CONTU stated that although both involve the creation of a derivative work, unless the reader or user "tries to copy and vend that work, the copyright owner is unlikely to be very concerned."²⁵⁴ Copyright owners having a different view "could, of course, make such desires a contractual matter."²⁵⁵

This last remark appears merely to refer to a copyright owner's right to contractually prohibit adaptations of the copyrighted work. However, some commentators have argued that CONTU meant to give copyright owners the right to contractually extinguish all of the rights that CONTU's proposed § 117 would give to rightful possessors of a copy of a program.²⁵⁶ If so, CONTU chose singularly poor wording for the proposed § 117, which would have created a potential trap for contract drafters. As the CONTU report made clear, a "rightful possessor" was to include not only a purchaser, but also a lessee or licensee of a copy of a computer program.²⁵⁷ Had § 117 been worded so as to permit all rightful possessors to make or authorize the making of another copy or an adaptation of the program, then it could have been argued that those rights could not be contractually extinguished unless the contract stated in the clearest possible terms that possession of the copy of the program being transferred would be rightful only so long as no copies or adaptations of the program were made. Even then, the courts might have construed the statutory rights of a "rightful possessor" to preempt enforcement of such a contract.²⁵⁸

Without explaining its motivation, Congress, in the 1980 amendments, changed the holder of § 117 rights from a "rightful possessor" to the "owner" of a copy of the computer program. Congress apparently intended by this change to allow copyright holders to contractually modify the rights enumerated in § 117 merely by retaining ownership of copies of a computer program and transferring possession of those copies by means of a lease or license restricting the uses to which the copies could be put.

2. Shrink-wrap Licenses and Other Contractual Restrictions on Reverse Engineering of Computer Programs

The importance of the change in the wording of § 117 has not been lost on the software industry. Developers of mass-distributed software now commonly print extensive contractual terms on software packaging and specify that removal of the wrapping by the user constitutes acceptance of those terms.²⁵⁹ These so-called "shrink-wrap" licenses often contain clauses prohibiting any modification or disassembly of the computer program or its incorporation into any other program.²⁶⁰ The terms may also specify that the software proprietor retains title to the software or diskette and that the user is merely a bailee or lessee rather than the owner.²⁶¹

Although several courts and commentators have questioned whether shrink-wrap agreements constitute enforceable contracts,²⁶² software developers continue to rely on them to deter reverse engineering. In an effort to buttress the enforceability of such provisions, software developers have persuaded two state legislatures, in Louisiana and Illinois, to enact statutes specifically providing that if certain conditions are met these shrink-wrap licenses are valid contracts.

The Louisiana and Illinois Software License Enforcement Acts²⁶³ specify that any person who acquires computer software will be conclusively deemed to have accepted and agreed to all terms of a license agreement for the software if a written notice of such terms is affixed to or packaged with the software in such a way as to be conspicuous upon casual examination, readily understandable, and clear in stating that use of the software or opening of the package will constitute acceptance of the license terms and that unused and unopened software may be returned for a full refund. The statutes also state that the terms deemed accepted in such cases can include any or all of the following: (1) Retention by the licensor of title to the copy of the software; (2) prohibitions or limitations on copying software or using copies made; (3) prohibitions or limitations on adapting the software in any way, including limitations on translating, reverse engineering, decompiling, or disassembling the software; (4) prohibitions or limitations on further transfers, rental or other disposition of the software or copies of same.²⁶⁴

However, reliance on shrink-wrap licenses as a way of limiting a user's rights under § 117 of the Copyright Act was called into question by the first court of appeals decision to consider § 117.

3. The Vault Decision

*Vault Corporation v. Quaid Software Limited*²⁶⁵ held that the Louisiana Software License Enforcement Act was preempted by federal copyright law. In addition, the court held that a routine used to defeat a copy-protection program was non-infringing.²⁶⁶

In *Vault*, the Court of Appeals for the Fifth Circuit affirmed a lower court holding that a feature of Quaid's "CopyWrite" diskette, called "RAMKEY," (which was developed, as it will be recalled, by loading Vault's copy-protected "PROLOK" diskette into a computer for analysis, and which, as developed, unlocked the "PROLOK" copy-protection program) neither directly infringed the copyright of the "PROLOK" copy-protection program nor contributed to "CopyWrite" users' infringement of the copy-protection program or programs placed on "PROLOK" diskettes by Vault's customers.²⁶⁷ The "RAMKEY" feature of Quaid's "CopyWrite" program was said to enable purchasers of programs on "PROLOK"-protected diskettes to make archival copies, as authorized by § 117 (2), that would protect them not only against mechanical or electrical failure but also against physical destruction of the "PROLOK"-protected diskette itself.²⁶⁸

The court of appeals also upheld the lower court's rejection of a claim that Quaid breached a "shrink-wrap" license agreement that accompanied the "PROLOK" diskettes. The lower court had reasoned that because the license agreement was a contract of adhesion, it was enforceable only if the Louisiana Software License Enforcement Act was valid and enforceable. The lower court held that because the Louisiana Act "touches upon an area" of federal copyright law, its provisions were preempted and Vault's license was unenforceable.²⁶⁹

A purchaser of a program on a "PROLOK" diskette could apparently make an archival copy of the program on another disk, but Vault's copy-protection device prevents a computer from reading the copy of the program unless the original "PROLOK" diskette is also in one of the computer's disk drives. The Court of Appeals went on to explain that:

The fact that a fully functional copy of a program cannot be made from a PROLOK diskette prevents purchasers from buying a single copy and making unauthorized copies for distribution to others. . . . Quaid's product . . . unlocks the PROLOK protective device and facilitates the creation of a fully functional copy of a program placed on a PROLOK diskette. The process is performed simply by copying the contents of the PROLOK diskette onto the CopyWrite diskette which can then be used to run the software program without the original PROLOK diskette in a computer disk drive. RAMKEY interacts with Vault's program to make it appear to the computer that the CopyWrite diskette contains the required "fingerprint," thereby making the computer function as if the original PROLOK diskette is in its disk drive.²⁷⁰

The court of appeals found no direct infringement in Quaid's reverse engineering of Vault's program because Quaid's act of loading a PROLOK diskette into its own computer memory for the purpose of analysis was "an essential step in the utilization of the computer program in conjunction with a machine," pursuant to § 117(1).²⁷¹ The program could be analyzed on the computer alone, and thus no decompilation, with its resulting "hard" (i.e. paper) copy of the program, was necessary.²⁷² The court declined to limit the qualifying phrase of § 117(1), requiring that the copy made by loading the program into the computer be "used in no other manner," to permit the user to load a program only for the purpose intended by its developer.²⁷³

Nor did the court find that Quaid's two versions of its "RAMKEY" feature constituted infringing derivative works based on Vault's program. The court held that to constitute a derivative work, "the infringing work must incorporate in some form a portion of the copyrighted work," and "must be substantially similar to the copyrighted work."²⁷⁴ Although one version of "RAMKEY" contained approximately 30 characters of source code copied from Vault's program, the court concluded that the copying was not significant and that in any event, the two versions of "RAMKEY" were not substantially similar to Vault's program. The court distinguished the cases on which Vault relied as involving derivative works that performed substantially the same function as the copyrighted work, whereas Quaid's programs performed an opposing function. The court also rejected Vault's contention that Quaid's program could be a derivative work if it altered the operation of Vault's copyrighted work.²⁷⁵ The case on which Vault relied, noted the court, merely held that the sale of a program that sped up the operation of another program constituted contributory infringement because the sped up programs were themselves derivative works.²⁷⁶

The court found no contributory infringement by Quaid because the Supreme Court's decision in *Sony Corporation of America v. Universal City Studios* limits contributory infringement to cases where the material or device involved is not capable of substantial non-infringing uses,²⁷⁷ and the "RAMKEY" feature of the "CopyWrite" program enabled purchasers of programs on "PROLOK"-protected diskettes to make archival copies, as authorized by § 117(2), that would protect them not only against mechanical or electrical failure (which could be avoided by copying a "PROLOK"-protected diskette on another diskette for use with the original "PROLOK"-protected diskette), but also against physical destruction of the "PROLOK"-protected diskette itself.²⁷⁸

Finally, the court of appeals held that the provision in the Louisiana Software License Enforcement Act permitting a software producer to prohibit decompilation or disassembly of a copy of its computer program "conflicts with the rights of computer program owners under § 117 and thus clearly 'touches upon an area' of federal copyright law."²⁷⁹ For that reason, the provision was preempted by federal law.²⁸⁰ This rendered the restriction in Vault's license agreement against reverse engineering unenforceable.

4. CRITICISMS OF THE VAULT DECISION

The *Vault* decision has been criticized as being a "rogue elephant" among those construing § 117.²⁸¹ There are four stated criticisms of the case. The first is that it ignored the fact that § 117 grants rights only to an "owner" of a copy of a computer program, whereas Quaid was merely a licensee.²⁸² As the leading critic of the *Vault* decision put it, the court

held that a statute dealing with licenses was preempted (and that Vault's license was therefore unenforceable) because it conflicted with a federal statute granting certain rights to owners. Thus, it cannot be said that the court moved Quaid into the category of § 117 owners because Vault's license was unenforceable; on the contrary, the license was unenforceable because the court treated Quaid as an owner. The only plausible inference is that the court simply ignored the distinction between owners and licensees. In so doing, it created ammunition for the argument that not only shrink-wrap licenses but all "rightful possessors" of software—the words of the statute as drafted by CONTU—are entitled to the benefits of § 117.²⁸³

The court's decision thus flew in the face of Congress's apparent intent in changing the words from a "rightful possessor" to "the owner" of a copy of computer software.

The second criticism of *Vault* is that it nullified one of CONTU's own premises—that copyright owners desiring more protection than was available under § 117 "could, of course, make such desires a contractual matter."²⁸⁴ The *Vault* decision would arguably preempt any contract that restricted reverse analysis and not just shrink-wrap licenses. Although the court in *Vault* merely held that the shrink-wrap licenses authorized by the Louisiana License Act were preempted because they contained restrictions more onerous than those of § 117, the same could be said of any contract, whether a shrink-wrap license or not, insofar as it restricts the uses of computer programs that § 117 would permit. Indeed, one subsequent federal trial court, following *Vault's* lead, may have held just exactly that.²⁸⁵

The third criticism of *Vault* is that the court should have construed the phrase "essential step in the utilization of a computer" as

permitting the copying of a computer program only for the purpose intended by the copyright holder.²⁸⁶ The court's conclusion that there is no language suggesting that the copy that § 117 permits to be made in loading a program into a computer must be employed for a purpose intended by the copyright owner, is "clearly at odds with the purpose of the section as stated by CONTU."²⁸⁷ The CONTU report itself states that inclusion of an adaptation right in § 117 was designed to provide "a right to make those changes necessary to enable the use for which [the program] was both sold and purchased."²⁸⁸

The fourth criticism of *Vault* is with respect to its holding that Quaid was not liable for contributory infringement because its program had a substantial non-infringing use in enabling the users of "PROLOK" diskettes to make archival copies.²⁸⁹ As we have seen, a purchaser of a program on a "PROLOK" diskette could make an archival copy as long as the original "PROLOK" diskette was also in one of the computer's disk drives.²⁹⁰ In contrast to previous district court opinions,²⁹¹ the court in *Vault* concluded that § 117(2) authorizes an owner of a copy of a computer program "to make an archival copy of that program in order to guard against all types of risks, including physical and human mishap as well as mechanical or electrical failure," and dismissed CONTU's reference to "mechanical or electrical failure" as "illustrative only, and not exclusive."²⁹² However, unlike mechanical or electrical failure, the risk of physical destruction is in no way unique to computer programs, and it is clear in other copyright contexts that the purchaser of a copy of a copyrighted work, such as a book, is not privileged to make a backup copy to guard against the risk of physical destruction, at least so long as additional copies of the book are available on the open market.²⁹³

Given these criticisms of the *Vault* decision, it is not surprising that the court of appeals in *Sega* declined to find that Accolade's reverse engineering was authorized by § 117.²⁹⁴ The *Sega* court concluded that Accolade's disassembly of the object code of Sega's program, conversion of the program into source code, and printouts and photocopies of the source code version went far beyond the uses contemplated by CONTU and authorized by § 117.²⁹⁵ Thus, the court found it unnecessary to decide whether § 117 permits a program to be used only as intended by the copyright owner, as Sega claimed, or permits the type of reverse analysis employed by Quaid in the *Vault* case.

As compelling as the foregoing criticisms of the *Vault* decision are, the recent *Atari* and *Sega* decisions suggest two further questions that the *Vault* court should have addressed: 1) Whether, irrespective of § 117, Quaid could have defended its reverse engineering as a fair use under § 107, just as the defendants in *Atari* and *Sega* succeeded in doing; and 2) whether the Louisiana Software License Enforcement Act could be said to conflict with and thus be preempted by § 107's fair use privilege. The first question will be discussed in the remainder of this section while the second question will be discussed in Part V.

With respect to the first question, the immediate purpose of Quaid's reverse engineering, as in *Atari* and *Sega*, was to gain access to the unprotected ideas and functional expression contained in the reverse engineered program. The ultimate purpose of Quaid's reverse engineering, however, was to produce a program that would defeat Vault's "PROLOK" program, whose particular "nature" was to prevent copying. Unlike the lockout devices in *Atari* and *Sega*, Vault's copy protection program was consistent with the goals of the Copyright Act, and was not a misuse. Thus, Quaid's ultimate purpose in reverse engineering Vault's program would undercut, rather than strengthen, any fair use defense.

Specifically, the product of Quaid's reverse engineering, namely, the "Copywrite" disk with its "RAMKEY" feature, could easily be said to contribute to others' infringement of Vault's "PROLOK" program. First, Vault's copy-protection program, as well as Vault's customers' programs, had to be copied onto the "Copywrite" diskette in order for the "RAMKEY" feature to work.²⁹⁶ This copy of the copy-protection program, unlike the copy made when the "PROLOK" diskette was loaded into the computer for reverse analysis, could not be characterized under § 117 as an essential step in the utilization of the "PROLOK" copy protection program in conjunction with a machine. Rather, the copy is produced as an essential step in the utilization of the "RAMKEY" feature of the "Copywrite" program, and thus constituted an infringing copy of the "PROLOK" program. Second, the argument that there was a "substantial non-infringing use" in assisting users of "PROLOK"-protected programs to make archival copies of Vault's customers' programs is less persuasive when we consider that the "RAMKEY" feature was not the only means available to avoid the risks caused by mechanical or electrical failure.²⁹⁷ Nor, as we have seen, could "RAMKEY" legitimately be used to avoid the risk of physical damage to "PROLOK"-protected programs.²⁹⁸ The "RAMKEY" feature had no apparent use other than to contribute to users' infringement of "PROLOK"-protected computer programs—a use which was highly likely, in turn, to diminish the market value of the "PROLOK" program.²⁹⁹

The ultimate unfairness of Quaid's conduct, however, flows not from the reverse engineering itself, but from the use made of the results of that reverse engineering. It is the purpose and character of the ultimate use, analyzed in accordance with § 107 of the Copyright Act, that make Quaid an infringer of Vault's program. Thus, notwithstanding the misguided result in *Vault*, the court's conclusion that reverse analysis that does not involve decompilation is not an infringement may have been correct.

As the discussion up to this point makes clear, although some uses of the product of reverse engineering of publicly distributed computer programs (such as defeating copy protection, as in *Vault*) may be unfair under § 107, other uses of the product of reverse

engineering (such as defeating lockout devices, achieving interoperability generally, or even developing a competing program) may nevertheless be fair. Reverse analysis would appear to be permitted under § 107 whether or not it is permitted under § 117.³⁰⁰ Any effort to enforce a contract purporting to limit the more general § 107 privilege to engage in decompilation of a publicly distributed computer program may constitute copyright misuse on the part of the copyright owner,³⁰¹ and, in any event, as will be seen in the next part of this article, is probably preempted by federal copyright and/or federal patent law.

V. Copyright and Patent Preemption of Restrictions on Reverse Engineering

A. Copyright Preemption

An appropriate place to begin a discussion of copyright preemption is to identify yet another criticism of the *Vault* decision, this one having to do with its copyright preemption analysis. Not only did the *Vault* court incorrectly find a conflict between the Louisiana Act and § 117 of the Copyright Act, but also, in applying the preemption test stated in *Sears, Roebuck & Co. v. Stiffel*³⁰² and its companion case, *Compco Corp. v. Day-Brite Lighting*³⁰³ (generally referred to collectively as the *Sears-Compco* cases), the court simply ignored the statutory preemption provision contained in § 301 of the 1976 Copyright Act.

Section 301(a) of the Act states that on or after the effective date of the Act,

all . . . 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 . . . are governed exclusively by this title.³⁰⁴

Section 301(b) goes on to state that nothing in the Copyright Act annuls or limits any rights or remedies under the common law or statutes of any State with respect to "activities violating . . . rights that are not equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106 . . ."³⁰⁵ Thus, the question is whether state statutory and contractual protections against reverse engineering fall within the scope of rights specified by § 106. An earlier version of § 301(b) contained a list of state law rights that were not considered equivalent to those specified in § 106. One such example was "breaches of contract."³⁰⁶ Although the clause was later deleted because of concerns over one of the other examples,³⁰⁷ the earlier version of § 301(b) nevertheless seems to have manifested a legislative intent not to preempt the state law of contracts.

Given the inclusion of § 301 in the Copyright Act, it could be argued that this express statutory preemption test, rather than the test quoted in *Sears-Compco*, should have been applied in *Vault*. The *Sears-Compco* cases, after all, were specifically concerned with the preemptive effect of federal patent law which, unlike the Copyright Act of 1976, contains no express statutory preemption provision. The test applied in *Sears-Compco* is arguably simply a rule for determining whether, in the absence of an express federal statutory preemption provision, state law nevertheless conflicts with federal law and is thus preempted by virtue of the Supremacy Clause of the United States Constitution.³⁰⁸

Of course, the inclusion of an express statutory preemption provision in the 1976 Copyright Act does not necessarily preclude a finding of preemption under the implied Supremacy Clause test articulated in *Sears-Compco*. A court could conceivably find that state law, although not expressly preempted by § 301(a), nevertheless "touches upon an area" governed by the Act, and does so in a such a way that federal policy "is set at naught, or its benefits denied."³⁰⁹ Such a conclusion, however, would require some explanation of the relationship between § 301(a) and the *Sears-Compco* preemption test. The court in *Vault* simply failed to provide such an explanation.

The legislative history makes clear that the primary purpose of § 301 was to preempt the common-law copyright protection for unpublished works, which coexisted with the federal statutory copyright protection for published works prior to the enactment of the 1976 Copyright Act.³¹⁰ The legislative history also states, however, that "[a]s long as a work fits within one of the general subject matter categories of §§ 102 and 103, [§ 301(a)] prevents States from protecting it even if it fails to achieve Federal statutory copyright protection."³¹¹ This remark suggests that § 301(a) itself might preempt any effort to provide contractual protection against reverse engineering of a publicly distributed computer program, where the program would fail to achieve federal copyright protection against such reverse engineering because of the fair use provisions of § 107. The problem with this conclusion is that contractual restrictions against reverse engineering are not "equivalent" to any of the exclusive rights contained in § 106. However, shrink-wrap licenses, which restrict virtually the entire public from reverse engineering publicly distributed computer programs, arguably do provide protection that is equivalent to copyright.

In any event, the legislative history also states that the purpose of § 301(b) "is to make clear, *consistent with the 1964 Supreme Court decisions* [in the *Sears-Compco* cases], that preemption does not extend to causes of action or subject matter outside the scope of the

revised Federal copyright statute"³¹² [emphasis added]. This statement provides authority for applying the *Sears-Compco* preemption test to state shrink-wrap licensing laws and other contractual restrictions on reverse engineering, even if § 301 itself does not preempt such provisions.

Although no court has yet ruled on whether § 107 would preempt enforcement of contracts restricting reverse engineering of computer programs, the Court of Appeals for the Second Circuit in *Wright v. Warner Books*³¹³ addressed the preemptive effect of § 107 with respect to contractual restrictions on the right to make use of unpublished letters. The court in *Wright* refused to construe a restrictive agreement in such a way as to prohibit a biographer from using manuscripts in a university archive for scholarly purposes (in the case at bar, for the writing of a biography of the author, Richard Wright). Purely as a matter of contract construction, the court said: "It defies common sense to construe this agreement as giving scholars access to manuscripts with one hand but then prohibiting them from using the manuscripts in any meaningful way with the other."³¹⁴ But the court went on: "To read [the restrictions agreed upon] as absolutely forbidding any quotation, no matter how limited or appropriate, would severely inhibit proper, lawful scholarly use and place an arbitrary power in the hands of the copyright owner going far beyond the protection provided by law."³¹⁵ This latter remark strongly suggests that the court would preempt enforcement of a contract that attempted to restrict § 107's fair use privilege.

Preemption by federal copyright law of contracts prohibiting reverse engineering appears consistent with the European Community Directive on the Legal Protection of Computer Programs. The EC Directive makes it clear that the right to reverse engineer copyrighted software, which is guaranteed by Articles 5(3) and 6, may not be withdrawn by contractual provisions. According to Article 9(1), any contractual provision contrary to these articles is void.³¹⁶

B. Patent Preemption of State Law and Primacy over Copyright Law

A final criticism of the *Vault* decision is that in applying the *Sears-Compco* rule, the court neglected a number of Supreme Court preemption decisions in which the Court refused to find preemption because it gave deference to state contract and trade secret law. However, as this section will indicate, the most recent Supreme Court patent preemption decision, *Bonito Boats, Inc. v. Thunder Craft Boats*,³¹⁷ supports the *Vault* decision with respect to preemption of the Louisiana shrink-wrap licensing law.

In *Bonito Boats*, the Supreme Court unanimously held that a Florida statute,³¹⁸ which prohibited the unauthorized use of a direct molding process to duplicate manufactured boat hulls, conflicted with federal patent law and was thus invalid under the Supremacy Clause. In the course of its opinion, the Court reviewed and reaffirmed its holding in the *Sears-Compco* cases, but also reaffirmed intervening decisions that refused to preempt various aspects of state trade secret and contract law.

In *Sears-Compco*, the Court had held that because of the federal patent laws, a state may not, through its unfair competition laws, prohibit the copying of an unpatented and uncopyrighted article. In both *Sears* and *Compco*, litigants had sought to enjoin the copying of publicly distributed articles whose patents had been declared invalid.³¹⁹ The Court concluded that to forbid copying would interfere with the federal policy of allowing free copying of whatever the federal patent and copyright laws leave in the public domain.³²⁰

Likewise, in *Brullotte v. Thys Company*,³²¹ the Court held that a patentee's use of a royalty agreement that projects the obligation to pay royalties beyond the expiration date of the patent is unlawful *per se*.³²² The Court concluded that if such a device were available to patentees, the free market visualized for the post-expiration period would be subject to monopoly influences.³²³ Similarly, in *Lear, Inc. v. Adkins*,³²⁴ the Court held that a patent licensee who establishes that the patent is invalid need not pay royalties accrued subsequent to the issuance of the patent.³²⁵ Both of these cases stood for the proposition that contracts which attempt to provide protection unavailable under federal patent law are preempted.

In *Kewanee Oil Co. v. Bicron Corporation*,³²⁶ the Court held that because state trade secret law has long co-existed with federal patent law and does not undermine federal patent policy, even where protection is sought for trade secrets that are clearly patentable, there is no conflict requiring preemption.³²⁷ Although not mentioned in *Kewanee Oil*, it is noteworthy that the patent statute itself, in specifying that applications for patents are to be kept confidential until a patent issues, embodies a policy of co-existence with state trade secret law.³²⁸ In *Kewanee Oil*, the Court reasoned that the danger of interference with federal patent law was remote, because state trade secret law provides far weaker protection than patent law, as the public at large remains free to discover and exploit the trade secret through reverse engineering of products in the public domain.³²⁹

Likewise, in *Aronson v. Quick Point Pencil Co.*,³³⁰ the Court held that preemption did not preclude enforcement of a contract to pay reduced royalties to a patent applicant after a denial of the patent application.³³¹ The Court concluded that enforcement of the royalty agreement was consistent with its decision in *Brullotte v. Thys Company*, because the reduced royalty being challenged, far from being negotiated "with the leverage" of a patent, rested on the contingency that no patent would issue within five years.³³² A concurring

opinion in *Aronson* added that nothing in *Brulotte* would justify preventing a patent applicant/licensor from entering into a contract whose term does not end if the application fails.³³³

Notwithstanding its pragmatic approach in *Kewanee* and *Aronson* to state trade secret and contract law, the Court in *Bonito Boats* reaffirmed *Sears-Compco*, at the heart of which, the Court said, "is the conclusion that the efficient operation of the federal patent system depends upon substantially free trade in publicly known, unpatented design and utilitarian conceptions."³³⁴ Although the Court admitted that its decisions since *Sears* "have taken a decidedly less rigid view of the scope of federal preemption under the patent laws,"³³⁵ it went on to state its belief "that the *Sears* Court correctly concluded that the States may not offer patent-like protection to intellectual creations which would otherwise remain unprotected as a matter of federal law."³³⁶

The Court in *Bonito Boats* stated:

In essence, the Florida law prohibits the entire public from engaging in a form of reverse engineering of a product in the public domain. This is clearly one of the rights vested in the federal patent holder, but has never been a part of state protection under the law of unfair competition or trade secrets.³³⁷

The Court went on to reiterate what it had noted in *Kewanee Oil*, that "the competitive reality of reverse engineering may act as a spur to the inventor, creating an incentive to develop inventions which meet the rigorous requirements of patentability."³³⁸

As we will see, the Supreme Court's decision in *Bonito Boats* may bear upon two separate aspects of the question of the legitimacy of reverse engineering.

1. Patent Preemption of State Shrink-Wrap Licensing Legislation

First, the *Bonito Boats* decision strongly suggests that federal patent law is an alternative to federal copyright law for preempting both state legislation and judicial cases that uphold the enforceability of "shrink-wrap" licenses, where the licenses prohibit virtually the entire public from reverse engineering a publicly distributed computer program. Although the *Vault* decision may have given erroneous reasons for why the Louisiana statute was preempted by federal copyright law, it did not address the question whether the Louisiana Act might be preempted by federal patent law. Some commentators, even before the *Bonito Boats* decision, concluded that federal patent law would indeed preempt such legislation.³³⁹

If shrink-wrap license provisions prohibiting reverse engineering are unenforceable because of the preemptive effect of federal copyright or patent law, then many current "rightful possessors" of copies of mass-marketed computer programs would perforce become *de facto* "owners."³⁴⁰ In the absence of the shrink-wrap provisions, owners would be free of any contractual limitations on their right under § 117 of the Copyright Act to load the programs into their computers, to adapt them as necessary for use on their computers, and to make archival copies. Courts could also consider some of the other statutory issues that the *Vault* decision raised concerning the permissibility, under § 117, of reverse analysis of a program after a lawfully obtained copy is loaded into a computer. At least where reverse engineering can be accomplished entirely within the computer, as in *Vault*,³⁴¹ it could be argued that the act of loading the program into the computer is indeed an "essential step"³⁴² in using the computer program. If this construction of § 117 is adopted, reverse analysis not involving decompilation or disassembly would thus be fair *per se*, without any need to consider the indispensability of the reverse engineering under § 107's four factor test, although the fairness of the use to which the product of that reverse engineering is put would still be subject to a § 107 fair use analysis. Such an interpretation of § 117 would be consistent with Article 5(3) of the European Community Directive.³⁴³

2. Patent Law Primacy over Copyright Law

The *Bonito Boats* decision may have a second, and more far-reaching effect on the reverse engineering question. Some commentators have suggested that the decision "may shed (perhaps no more than a glimmer of) light on the Court's view of the relationship, not just between [federal] patent law and state law, but also between patent and copyright law as they affect computer software."³⁴⁴ This is because "[t]he patent system is traditionally thought of as (what may be described as) 'preempting' copyright law in certain areas, just as it preempts state law in some respects."³⁴⁵ Although the primacy of patent law over copyright "is usually discussed in terms of statutory construction," because patent and copyright law are co-equal bodies of federal law, nevertheless, "analyzing that primacy as a species of preemption can provide a useful perspective."³⁴⁶ That perspective is legitimate, commentators argue, because "the Copyright Act contains its own self-'preempting' provision: the statute [in § 102(b)] provides that copyright does not extend to any method of operation, process, or procedure, among other things."³⁴⁷

Thus far, the principal conclusion of the commentators offering the foregoing view is that it "may call into question the vitality of the *Whelan* line of cases holding that the structure, sequence and organization of computer software-i.e., important functional characteristics of the software-are protectible by copyright."³⁴⁸ The question raised by this line of cases, as we have seen, has an important, albeit indirect, bearing on the reverse engineering question, in that narrowing the scope of copyright protection for computer programs strengthens the justifications for reverse engineering as a necessary means for obtaining access to the broad universe of unprotected ideas.

But *Bonito Boats* may have an even more direct bearing on the reverse engineering question. The case also buttresses the conclusions of the courts in *Atari*, *Sega* and *Vault*-namely that reverse engineering of a publicly distributed program is a fair use under § 107, and that a person who loads a program into a computer in accordance with § 117 may thereafter subject the program to reverse analysis.

The argument for patent preemption of state law and patent primacy over federal copyright law gains strength from the increasing recognition in the United States that computer programs may indeed be the subject of patent protection. Before 1981, the patentability of computer programs was in considerable doubt. In that year the Supreme Court made clear in its decision in *Diamond v. Diehr* that "a claim drawn to subject matter otherwise statutory [i.e., patentable] does not become non-statutory simply because it uses a mathematical formula, computer program or digital computer."³⁴⁹ The Court held that a claim containing a mathematical formula used in a computer program must be viewed as a whole, and patent protection will issue if the overall function of the formula is to "transform or reduce an article to a different state or thing."³⁵⁰ This language generally confirms earlier decisions of the Court of Customs and Patent Appeals (CCPA) holding that the application of a formula either to physical elements (in apparatus claims) or to process steps (in process claims) was sufficient to establish the validity of a patent claim,³⁵¹ although the specific holding in *Diehr* merely identified the physical transformation of an article as the touchstone of validity for a claim containing a mathematical formula.³⁵²

Subsequent decisions by lower courts have held that computer programs are proper subject matter for patent protection. Several post-*Diehr* decisions by the CCPA are of particular relevance. *In re Taner*,³⁵³ for example, held that a method for converting electrical signals, by a process that includes a mathematical formula, is patentable subject matter.³⁵⁴ *In re Pardo*³⁵⁵ upheld claims for a method and apparatus for controlling the internal operations of a computer so as to convert it from a sequential processor to one not dependent upon the order in which it receives program instructions.³⁵⁶ A federal trial court decision, *Paine, Webber v. Merrill Lynch*,³⁵⁷ upheld a patent for a computer program dealing with cash management, stating that it claimed a "methodology to effectuate a highly efficient business system" rather than a mere mathematical formula.³⁵⁸ Finally, the court of appeals for the federal circuit has upheld the validity of computer-program-related claims.³⁵⁹

As the foregoing cases illustrate, and a recent survey confirms, the Patent and Trademark Office is in fact issuing "pure" software patents in significant numbers.³⁶⁰ The survey defines pure software patents as "patents which specifically disclose and claim software technology without referring to hardware, other than a computer and typical peripheral devices."³⁶¹ These patents can be divided into two general types-inventions related to business methods and those related to computer/user interfaces.³⁶² The survey concludes that:

[T]he Patent Office is now issuing a large number of patents for computer programs operating on computers in a wide variety of applications. Many of these patents are "pure" software patents which indicates that the Patent Office is now willing to grant patents for novel and non-obvious computer programs operating on conventional, off-the-shelf computer hardware.³⁶³

The Patent Office and the federal courts now face what may be thought of as a second generation of computer software patent cases, in which the concern will shift to the novelty and non-obviousness requirements as applied to computer programs, and the degree of disclosure required of the actual source code of a program.

The first paragraph of § 112 of the federal patent laws contains three disclosure requirements for a patent specification: 1) a written description of the invention; 2) a disclosure of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most closely connected, to make and use the invention (the enabling disclosure); and 3) a disclosure of the best mode contemplated by the inventor for carrying out the invention.³⁶⁴

A recent review of the cases dealing with what level of disclosure of software related inventions is necessary to satisfy these disclosure requirements concludes that non-disclosure of computer codes in a patent application is not a *per se* violation of either the enablement or best mode requirements of § 112, and that the application need only disclose sufficient information to ensure that a person of ordinary skill in the art could practice the invention without undue experimentation and could routinely write the best mode

program.³⁶⁵ This same article also concludes, however, that maintaining the computer code as a trade secret could give rise to an inference that the disclosure is inadequate, and that where computer code is not disclosed, objective evidence in the form of expert testimony and affidavits may be necessary to withstand a § 112 challenge to the adequacy of disclosure.³⁶⁶ These disclosure requirements, it will be noted, stand in marked contrast with the Copyright Office regulations enabling those registering the source code version of a computer program to register only identifying portions of the program and, under some circumstances, to block out portions of the source code containing trade secret material.

As we have seen, the existence of these regulations contributes in a material way to the § 107 fair use argument that the nature of computer programs is such that reverse engineering is the only way to gain access to the work's uncopyrighted ideas. Even if the copyright office regulations are found not to be inconsistent with the fair use provisions of the Copyright Act, their validity should be contingent upon the existence of a broad fair use privilege on the part of rightful possessors of publicly distributed object-code versions of a computer program to reverse engineer the program. Otherwise, the regulations would undermine what the Supreme Court in *Bonito Boats* described as the heart of its earlier decisions in *Sears-Compco* and *Kewanee Oil*—the conclusion that the efficient operation of the federal patent system depends upon substantially free trade in publicly known, unpatented design and utilitarian conceptions. As the Court in *Bonito Boats* said, the competitive reality of reverse engineering is not only in keeping with that objective, but may act as a spur to the inventor, creating an incentive to develop inventions which meet the rigorous requirements of patentability.

†. B.A., 1964, Birmingham Southern College; M.A., J.D., Duke University, 1972. Professor of Law, Washington University, St. Louis, Missouri, U.S.A. I would like to thank Professor Yoshiaki Nakamura, of Nihon University College of Economics, and Professor Tadakuni Nogimura, of Nihon University College of Law, for their generosity in arranging for me to deliver a series of lectures in Japan, in May 1991, out of which this article grew. I would also like to thank Nuno Carvalho, Cheryl Rose and Seth Phelps for their invaluable assistance in helping prepare this article for publication. An earlier version of this article appears in Nihon University's journal, *Comparative Law*, under the title, *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States*, 9 COMPARATIVE LAW 13 (1992).

1. Throughout this article, the term "computer program" will be used as it is defined in § 101 of the Copyright Act of 1976, 17 U.S.C. § 101 (1988), to refer to "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." *Id.* The term "software" will be used in this article as a more generic term that includes computer programs, databases, and documentation. *See generally* 1 DAVID BENDER, COMPUTER LAW §§ 1.05, 2.06[1] (1992).

2. For examples of judge-made law in each of these areas of intellectual property, see *infra* Part III. For a discussion of semiconductor chip design protection, see *infra* Parts III, IV(C)(2).

3. Council Directive 91/250 of 14 May 1991 on the Legal Protection of Computer Programs, O.J. (L122/42) [hereinafter EC Directive]. For a discussion of the E.C. Directive, see *infra* Parts IV(B)(C)(D). *See also* Keiji Sugiyama, *Reverse Engineering and Other Issues of Software Protection in Japan*, 11 EUR. INTELL. PROP.REP. 395 (1991) (observing that most Japanese scholars consider reverse engineering to be a lawful practice).

4. The Supreme Court of the United States has described reverse engineering as "starting with the known product and working backward to divine the process which aided its development or manufacture." *Kewanee Oil Corp. v. Bicron Corp.*, 416 U.S. 470, 476 (1974). As applied to computer programs, reverse engineering embraces a wide variety of methods for analyzing existing computer programs. For a detailed explanation of reverse engineering of computer programs and the difference between disassembly, decompilation, and other forms of reverse engineering, see *infra* note 15. For the reasons a computer program user or developer may wish to reverse engineer a program, see *infra* text accompanying note 19.

5. 975 F.2d 832 (Fed. Cir. 1992).

6. 977 F.2d 1510 (9th Cir. 1992), *amended*, 1993 U.S. App. LEXIS 78.

7. 17 U.S.C. §107 (1988).

8. *See, e.g., infra* Part III.

9. 847 F.2d 255 (5th Cir. 1988).

10. For discussion of the first question, see *infra* Part IV(C)(3)(a). For a discussion of the second, see *infra* Part V.

11. For an explanation of the difference between source code and object code, see *Apple Computer v. Franklin Computer Corp.*, 714 F.2d 1240, 1243 (3d Cir. 1983). For a discussion of trade secret protection for source code, see *infra* Part III(A).

12. See L.J. KUTTEN, *COMPUTER SOFTWARE* (1992) § 9.01, (quoting *In re Bedford Computer Corp.*, 62 Bankr. 555, 562 n.6 (C.D. N.H. 1986)) ("[T]he source code is the 'Life-blood' of any computer company and it is rarely disclosed."). Kuten discusses at length the escrow arrangements which have been used in the software industry to protect secrecy. See also Allen R. Grogan, *Decompilation and Disassembly: Undoing Software Protection*, *COMPUTER LAW*. Feb.1984, at 1, 3, 11 (noting that underlying design elements are viewed as protectible trade secrets because they determine such factors as the software's speed, accuracy, cost, and commercial feasibility, and concluding that through carefully drafted, enforceable agreements, a licensor should be able to restrict or prohibit reverse engineering, and that if programs are sold without restriction, any attempt to decompile or disassemble object code should be deemed to constitute copyright infringement unless it is undertaken 1) for non-profit or similar purposes and otherwise falls within the fair use provisions of section 107 of the 1976 Copyright Act, or 2) to use the program in conjunction with a machine as contemplated by section 117 of the Act.); Anne C. Keays, *Software Trade Secret Protection*, 4 *SOFTWARE L.J.* 577 (1991) (discussing trade secret protection under the Restatement (First) of Torts and the Uniform Trade Secrets Act, as well as the scope of trade secret protection in Europe and Japan); Ronald S. Laurie, *Protection of Trade Secrets in Object Form Software: The Case for Reverse Engineering*, *COMPUTER LAW.*, July 1984, at 1 (noting that many software producers consider the internal structure and logic of their software to be a trade secret despite unrestricted distribution of the object code form, but concluding that an act of decompilation or disassembly performed in order to gain access to the internal structure and logic of the program does not constitute copyright infringement, and that a trade secret claim based on decompilation or disassembly of object code, absent a contractual or confidential relationship, is preempted by federal copyright law). For a discussion of decompilation and disassembly, see *infra* note 15.

13. See Grogan *supra* note 12, at 2. See also Andrew Johnson-Laird, *Reverse Engineering of Software: Separating Legal Mythology from Actual Technology*, 5 *SOFTWARE L.J.* 331, 343 (April, 1992) ("Deciphering computer-executable programs is extremely tedious and error prone; it can take up to a minute or so for each computer instruction (a typical program might contain 500,000 instructions-347 days' worth of deciphering).").

14. Computer programming has been described as a three-step process that works from the general to the specific: 1) Defining the problem to be solved; 2) flow charting the logical sequence of steps to be performed in solving the problem and breaking the steps down into increasingly specific subroutines or modules; and 3) coding the program. The third step, in turn, involves two further steps: first, writing the program in human-readable source code (which can consist of either any one of several high-level computer languages that resemble human language and utilize human logic, or of assembly language, which requires the programmer to approach each problem using generally far more detailed machine logic); and second, utilizing an interpreter or compiler program for high-level source code, or an assembler program for assembly language source code, to translate the source code into machine-readable form, called object code. Object code is simply a binary language, consisting of zeros and ones, through which the computer hardware directly receives its instructions as a series of electronic pulses generated by the object code. See generally 1 BENDER, *supra* note 1, § 2.03; Laurie, *supra* note 12, at 3-4.

15. For an accessible non-technical description of the process of reverse engineering of computer programs and its role in software development, see Johnson-Laird, *supra* note 13. This commentator claims that the term "reverse engineering" is a misnomer, "a convenient 'handle' ascribed to the process of analyzing existing software." *Id.* at 342. Reverse engineering, he says, is actually an "additive" process, by which programmers must "recreate" the logical structure of the original program, which demands a considerable amount of information from the reverse engineer. *Id.* at 344. See also Thomas C. Vinje, *The Development of Interoperable Products Under the EC Software Directive*, *COMPUTER LAW.*, Nov. 1991, at 3. Vinje distinguishes between "reverse analysis" techniques, such as test runs, communication line traces, storage media dumps, and screen displays of object code, which do not involve translation of the analyzed program's object code into anything similar to its original assembler source code, and those techniques which do involve such a translation, known as disassembly or decompilation. *Id.* at 10 n.23. Disassembly and decompilation attempt to reverse the assembly and compilation processes described *supra* note 14. Of these two reverse engineering techniques, Vinje notes:

Disassembly is conducted by using a disassembler program to translate object code back into something akin to assembler source code. "Decompilation" has come to be used more generally to connote the computerized translation of a computer program's object code into something akin to its origin source code. . . . The appropriateness of the term "decompilation" has been questioned by many who deny the existence (or even the theoretical possibility) of any technique by which high level source code can be recreated from object code. In any event, disassembly is a technique

that clearly exists-indeed any operating system will include as a matter of course a disassembly tool-and disassembly has been subsumed within the term "decompilation."

Id. at 10 n.38. *See also* Grogan, *supra* note 12, at 7:

A compiled program is likely to be relatively more time-consuming and difficult to reverse engineer than a program written in assembly language . . . because the internal structure of a program which has been compiled and optimized tends to be jumbled and confusing. . . . [A] decompiler will not be able to generate the higher level language in which the compiled program was originally written.

Id. Reverse engineering of computer programs should not be confused with reverse engineering of semiconductor chips. For a discussion of the difference, see *infra* note 57.

16. *See* 1 BENDER, *supra* note 1, § 2.06[2].

17. *Id.*

18. *See* Ronald S. Laurie, *The Copyrightability of Microcode: Is It Software or Hardware . . . or Both?*, COMPUTER LAW., Mar. 1985, at 1.

19. The foregoing list of reasons for engaging in reverse engineering is based on a similar list contained in a report of the Committee on Computer Law, *Reverse Engineering and Intellectual Property Law*, 44 REC. ASS'N. B. CITY OF N.Y., 132, 134-35 (1989), (hereinafter "Committee Report"), *reprinted in* 3 ROGER MILGRIM, MILGRIM ON TRADE SECRETS (1991) App. B-8, *in* 12-12B BUSINESS ORGANIZATIONS (1990). For a discussion of this report, see *infra* Part IV(C)(3).

20. *See infra* notes 37-39, 45-51, and accompanying text, and Part V(B), discussing patent preemption of state law.

21. *See supra* note 12 and accompanying text.

22. *See infra* Part IV(C)(2) for a discussion of the Chip Act.

23. *See infra* notes 53-55 and accompanying text, and Part IV(C)(3)(a), discussing trademark misuse and the first fair use factor.

24. *See* BENDER, *supra* note 1, §3.02[4].

25. *See infra* Part III(B).

26. *See generally* 1 MILGRIM, *supra* note 19, §§ 2.03-.04, 3.01, 4.01.

27. *Id.* at §§ 5.04-05.

28. *Id.* at § 5.04[2]-[3].

29. *Id.* at § 5.04-05.

30. *Id.* at § 5.04[1].

31. Section 702 of the 1976 Copyright Act authorizes the Register of Copyrights "to establish regulations not inconsistent with law for the administration of the functions and duties made the responsibility of the Register under this title." 17 U.S.C. § 702 (1988). Section 705 of the 1976 Copyright Act specifies that articles deposited in connection with a completed registration are to be open to public inspection, but section 706 states that copies of deposited articles are to be authorized or furnished "only under the conditions specified by the Copyright Office regulations." 17 U.S.C. §§ 702, 706 (1988). These regulations, in turn, permit reproduction of deposited articles only if 1) the copyright owner grants permission, 2) a court orders reproduction, or 3) the Copyright Office receives a written request from an attorney on behalf of a party to actual or prospective litigation involving the copyrighted work, and the attorney gives

satisfactory assurances that the requested reproduction will be used only in connection with the specified litigation. 37 C.F.R. § 201.2 (d)(2)(i)-(iii) (1991).

32. 37 C.F.R. § 202.20(c)(2)(vi) (1992).

33. *Id.*

34. 37 C.F.R. § 202.20(c)(2)(vi)(vii) (1992).

35. Section 102 specifies that

(a) Copyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Works of authorship include the following categories:

(1) literary works;

(2) musical works, including any accompanying words;

(3) dramatic works, including any accompanying music;

(4) pantomimes and choreographic works;

(5) pictorial, graphic, and sculptural works;

(6) motion pictures and other audiovisual works;

(7) sound recordings; and

(8) architectural works.

(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.

17 U.S.C. § 102 (1988).

Section 101 defines "literary works" 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).

Apple Computer v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983), held that both the source code and object code of a computer program are protectible by copyright law, that the embodiment of a program in a read-only-memory (ROM) does not deprive a program of its copyright protection, and that operating programs as well as application programs are protected by copyright. *NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d 1177 (N.D. Cal. 1989) held, among other things, that microcode embedded in a microprocessor is a protectible computer program. For the court's observations on reverse engineering of microcode, see *infra* notes 128-30 and accompanying text. See generally Laurie, *supra* note 12, at 1 (classifying computer programs by form, function and medium of expression and discussing the copyrightability of microcode).

36. See generally 1 BENDER, *supra* note 1, § 4.04[2] (discussing the case law concerned with copyright protection for screen displays); Ross Katchman, *Copyright Registration of Computer Screen Displays from the Perspective of the Copyright Office*, COMPUTER LAW., Oct. 1987, at 16. For a polemic critique of the Copyright Office regulations governing registration of screen displays, see Gary L. Reback & David L. Hayes, *A Candid Assessment of the New Software Copyright Registration Procedures*, COMPUTER LAW., July 1989, at 1. Some courts have misleadingly referred to screen displays as non-literal elements of the program that generates them, while other courts have equally misleadingly suggested that a similarity between screen displays is evidence of a

similarity in the expression of the programs that generate them. See e.g., *Whelan Assoc. v. Jaslow Lab.*, 797 F.2d 1222, 1244 (3d Cir. 1986) (holding that screen displays could serve as indirect evidence of copying of the underlying program); *Broderbund Software v. Unison World*, 648 F.Supp 1127 (N.D. Cal. 1986) (treating screen displays as non-literal elements of the underlying computer program). Although Copyright Office regulations allow a single registration for computer programs and the screen displays they generate, the two are nevertheless conceptually distinct copyrightable works. Identical screen displays, moreover, can be generated by computer programs that do not themselves contain substantially similar expression.

37. See *Survey of U.S. Business Software Patents: Post-Diehr Through December 1990*, State Bar of Michigan, 56th Annual Meeting, reprinted in 1 BENDER, *supra* note 1, App. 3A[6]. An earlier version of the survey is reprinted in 1 BENDER, App. 3A[4]. The [earlier] version of the survey defines "pure" software patents as patents that "specifically disclose and claim software technology without directly referring to hardware, other than a conventional computer and peripheral devices." *Id.* at App. 3A-102. But see Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025 (1990). For a more extended discussion of patent protection for computer programs, see *infra* Part(V) (B).

38. See 1 BENDER, *supra* note 1, § 3A.03[2][c]. But cf. D.C. Toedt, *Software Patent Controversies Lead to Different Outcomes in the Federal Circuit, PTO*, COMPUTER LAW., July 1992, at 18. See generally Samuelson, *supra* note 37.

39. See Robert Barr & Susan Hollander, *Design Patents Revisited: Icons as Statutory Subject Matter*, COMPUTER LAW., June 1992, at 13, 15.

40. 17 U.S.C. §§ 901 *et seq.* (1988).

41. See RICHARD STERN, SEMICONDUCTOR CHIP PROTECTION § 1.1 (1986). Stern states that the two most important types of semiconductor chip product, in terms of sales volume, are microprocessors and memory chips. *Id.* at 4.

42. See 17 U.S.C. §§ 102, 107 (1988). For the text of sections 102 and 107, see *supra* note 35 and *infra* note 121, respectively.

43. The appropriation that a defendant has a right to prevent includes any use of the copyrighted work that violates the exclusive rights enumerated in 17 U.S.C. § 106 (1988). For the text of section 106, see *infra* note 60. For a discussion of the scope of protection for factual or utilitarian works, see *infra* notes 65-66 and accompanying text.

44. *Feist Publications, v. Rural Tel. Serv. Co.*, 111 S. Ct. 1282 (1992).

45. 35 U.S.C. §§ 103, 171 (1988) (inventions and designs, respectively).

46. See 1 BENDER, *supra* note 1, § 3A.02.

47. 35 U.S.C. §§ 112, 159 (1988).

48. In two respects, the disclosure of the invention benefits the public during the patent term. On the one hand, disclosure informs the public about the very existence of the invention and may trigger a process of negotiation with the patentee aimed at reaching a license agreement. On the other hand, disclosure, with the accompanying description of the state of the art, is helpful for competitors "to invent around" the potential invention when the invention is commercially successful and the patentee is not willing to license it. Contrary to the general belief, "inventing around" is not necessarily wasteful of social resources-on the contrary, it is procompetitive and stimulates innovation. It is precisely the procompetitiveness of "inventing around" in the patent context that led Congress, in section 906 of the Semiconductor or Chip Protection Act, 17 U.S.C. § 906 (1988), to authorize the reverse engineering of chips, and led the Ninth and Federal Circuits, in the *Atari* and *Sega* cases, to allow the reverse engineering of computer programs.

49. See 1 BENDER, *supra* note 1, § 3A.05; David Bender & Anthony Barkume, *Disclosure Requirements for Software-Related Patents*, COMPUTER LAW., Oct. 1991, at 1, 3. For a more detailed discussion of the disclosure requirements for computer program related inventions, see *infra* notes 364-66 and accompanying text.

50. 35 U.S.C. §§ 154, 271 (1991). As a result of the Patent Law Amendments Act of 1984, Pub. L. No. 98-622 § 101(a), 98 Stat. 3383 (1984), it is an infringement to supply within the United States, or to export from the United States, all or a substantial portion of the

uncombined components of a patented invention in such a manner as to actively induce a combination of such components outside the United States that would infringe the patent if such combination occurred within the United States. 35 U.S.C. § 271(f) (1988). This provision gives extraterritorial effect to U.S. patent law by making it an infringement to contribute to conduct occurring outside the United States that would be an infringement if it took place within the United States. As a result of provisions in the Omnibus Trade and Competitiveness Act of 1988, Pub.L. No. 100-418, § 9006, 102 Stat. 1156 (1988), it is an infringement to import into the United States, or to sell or use within the United States, a product which is made outside the country by a process patented in the United States 35 U.S.C. § 271(g) (1991). This provision gives extraterritorial effect to U.S. patents on processes.

51. The Patent Act explicitly acknowledges two species of patent "fair use." Section 271(e)(1) authorizes pharmaceutical laboratories to use patented substances (provided they do not result from genetic manipulation techniques) for the purpose of performing experiments and obtaining data in order to provide the FDA with the necessary information for the approval of a new drug. 35 U.S.C. § 271(e)(1) (1988). The new drug may be distributed only after the patent expires. *Id.* at § 271(e)(2). Section 272 makes it lawful to use patented inventions in any vessel, aircraft, or vehicle entering the United States temporarily or accidentally. 35 U.S.C. § 272 (1988). This section embodies the language of Article 5 of the Paris Convention for the Protection of Industrial Property, Mar. 20, 1883, 25 Stat. 1372, T.S. No. 379, *as revised by* Act of Stockholm, July 14, 1967, 21 U.S.T. 1629, T.I.A.S. No. 1583.

In addition to these two statutorily recognized fair uses, the unauthorized use of patented inventions for experimental purposes is generally considered a "fair use," under both U.S. case law and foreign patent statutes. *See, e.g.,* British Patents Act of 1977, Section 60/5; French Patent Law of 1968, as amended in 1978, Section 29; German Patent Law of 1980, Section 11; Japan Patent Law of 1959, as amended in 1978, Section 69; *Kaz Mfg. Co. v. Cheesebrough-Pond's, Inc.*, 211 F. Supp. 815, 818 (S.D.N.Y. 1962), *aff'd* 317 F.2d 679 (2d Cir. 1963). But the experimental use exception in the United States is narrowly defined. *See, e.g., Roche Products v. Bolar Chem. Co.*, 733 F.2d 858 (Fed. Cir. 1984) (unlicensed experiments, where carried out one year before a patent expired, not merely for amusement or strictly philosophical inquiry, but with a view to adapting the patented invention to the experimenter's business as soon as the patent expired, constitute infringement).

52. The copyright term for works in general is the life of the author plus fifty years. 17 U.S.C. § 302(a) (1988). For anonymous and pseudonymous works or works made for hire, the copyright term is seventy-five years from the year of its first publication, or one hundred years from its creation, whichever occurs first. 17 U.S.C. § 302(c) (1988). The term for utility patents is seventeen years, and the term for design patents is fourteen years. 35 U.S.C. §§ 154, 173 (1988) (utility and design, respectively).

53. *See, e.g., In re Morton-Norwich Products*, 671 F.2d 1332, 1336 (C.C.P.A. 1982). For a discussion of this limitation in federal copyright law, see *infra* notes 63-66, 70-71, 207-214 and accompanying text.

54. *See* 2 THOMAS MCCARTHY, MCCARTHY ON TRADEMARKS AND UNFAIR COMPETITION, §§ 23.01, 24.14 (3d ed. 1992).

55. Federal and state trademark law also recognizes a "fair use" defense for use of a descriptive term in its primary meaning. In some circumstances, a junior user has the "qualified right" to the use of a personal name. And, to some extent, a rebottler or repacker may "collaterally" use a trademark with the goal of keeping the public informed. *See* 1 & 2 MCCARTHY, *supra* note 54, §§ 11.17, 13.03, 25.08.

56. 17 U.S.C. § 904 (1988).

57. 17 U.S.C. § 906 (1988). For an illustration of the difference between reverse engineering of a semiconductor chip and reverse engineering of a computer program see *Atari Games Corp. v. Nintendo of Am.*, 975 F.2d 832, 836, (Fed Cir. 1992), where the court of appeals notes that Atari first tried to analyze Nintendo's lockout program by monitoring the communication between the "master" semiconductor chip in the console and the "slave" chip located in the game cartridge. *Id.* at 836. This is an example of reverse analysis of a computer program. Atari next tried to analyze the chips themselves by chemically peeling away layers of the chip to allow microscopic examination of the object code embedded therein. This is an example of reverse engineering of a semiconductor chip design. The latter process does not reverse engineer the embedded program, however, because it does not translate the embedded object code into source code. *Id.* To successfully reverse engineer the embedded program, Atari ultimately found it necessary to make false statements to the Copyright Office in order to obtain a copy of the source code for Nintendo's program.

The district court in *Sega Enterprises Ltd. v. Accolade, Inc.*, 785 F. Supp. 1392 (N.D. Cal. 1992) confused reverse engineering of a chip with reverse engineering of the computer program embedded therein. ("Accolade could have 'peeled' the microchips as set forth in § 906 . . . or programmed in a 'clean room,' but instead chose to disassemble, reproduce and enhance SEL's software."). *Id.*; *cf.* *Atari Games Corp. v. Nintendo of Am.*, 18 U.S.P.Q. 1935 (1991) (if Atari had proceeded with its analysis of "peeled" Nintendo chips as allowed by § 906, and not used the copy wrongfully obtained from the Copyright Office, it would not have infringed Nintendo's

copyright).

For a discussion of semiconductor chip protection and reverse engineering, see *infra* Part IV(C)(2).

58. For patent misuse, see, *e.g.*, *Brulotte v. Thys Co.*, 379 U.S. 29, 33 (1964), *Dawson Chemical Co. v. Rohm & Haas Co.*, 998 U.S. 176 (1980); *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488, 492 (1942); *Motion Picture Patents Co. v. Universal Film Mfg. Co.*, 243 U.S. 502 (1917). For copyright misuse, see, *e.g.*, *Lasercomb Am. v. Reynolds*, 911 F.2d 970, 973-74 (4th Cir. 1990); *M. Witmark & Sons v. Jensen*, 80 F. Supp. 843 (D. Minn. 1968), *aff'd per curiam*, 177 F.2d 515 (8th Cir. 1969). For trademark misuse, see, *e.g.*, *Anti-Monopoly, Inc. v. General Mills Fun Group*, 611 F.2d 296 (9th Cir. 1979). For a more detailed discussion of the misuse defense, see *infra* Part IV(C)(3)(a).

59. See *supra* note 35.

60. Section 106 states that:

Subject to sections 107 through 120, the owner of copyright under this title has the exclusive rights to do and to authorize any of the following:

(1) to reproduce the copyrighted work in copies or phonorecords;

(2) to prepare derivative works based upon the copyrighted work;

(3) to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending;

(4) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works, to perform the copyrighted work publicly; and

(5) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work, to display the copyrighted work publicly.

17 U.S.C. § 106 (1988). For a discussion of two provisions of the Act that arguably limit the exclusive rights as applied to computer programs, see *infra* Part IV(C) (§ 107), and Part IV(D) (§ 117).

61. See *generally* Laurie, *supra* note 12, at 1; Grogan, *supra* note 12, at 1; Committee Report, *supra* note 19.

62. 17 U.S.C. §102(b) (1988).

63. See *generally* MELVILLE B. NIMMER, 1 NIMMER ON COPYRIGHT § 2.03[D] (1992).

64. See *Computer Assoc. Int'l v. Altai, Inc.*, 982 F.2d 693, 707-708 (2d Cir. 1992), (citing *Concrete Mach. Co. v. Classic Lawn Ornaments*, 843 F.2d 600, 606 (1st Cir. 1988)); *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971); *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 66 (D. Mass. 1990); *Manufacturers Technologies v. Cams, Inc.*, 706 F. Supp. 984, 995-99 (D. Conn. 1989).

65. *Feist Publications v. Rural Telephone Service Co.*, 111 S. Ct. 1282 (1991) ("This Court has long recognized that the fact/expression dichotomy limits severely the scope of protection in fact-based works."); *Sega Enterprises Ltd. v. Accolade, Inc.*, 1993 U.S. App. LEXIS 78, *39-40 ("To the extent that a work is functional or factual, it may be copied. . . . Works of fiction receive greater protection than works that have strong factual elements . . . or works that have strong functional elements, such as accounting textbooks.") (citing *Baker v. Selden*, 101 U.S. 99, 104 (1879)).

66. *Sega*, 1993 U.S. App. LEXIS 78, at *48. ("In order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator of the work must satisfy the more stringent standards imposed by the patent laws."); *Atari*, 975 F.2d at 839 ("[P]atent and copyright laws protect distinct aspects of a computer program . . . Title 35 protects the process or method performed by a

computer program; Title 17 protects the expression of that process or method.").

Cf. § 101 of the 1976 Copyright Act, which states that pictorial, graphic or sculptural works, listed as copyrightable subject matter in section 102(a) of the Act, "include works of artistic craftsmanship insofar as their form but not their mechanical or utilitarian aspect are of concern," and that the design of a useful article "shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic or sculptural features that can be identified separately from, and are capable of existing independently of the utilitarian aspects of the article." 17 U.S.C. 101 (1988).

67. For a general discussion of the two generations of computer program copyright cases and the differing views of commentators over both generations of cases, compare Anthony L. Clapes et. al., *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 U.C.L.A. L. REV. 1493 (1987) (favoring broad copyright protection for computer programs), with Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045 (1989) and Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 JURIMETRICS J. 33 (1987) (opposing broad copyright protection for computer programs). *See also* John T. Soma, et al., *A Proposed Legal Advisor's Roadmap for Software Developers*, 68 DENV. U. L. REV. 191, 198-205 (1991) (describing three generations of computer copyright cases).

68. *See* Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 DUKE L.J. 663, 705-06.

69. *Id.* at 710.

70. *Id.* at 727.

71. *See id.* at 735.

72. *See* cases cited *supra* note 35.

73. *Compare* Whelan Assoc. v. Jaslow Dental Lab., 797 F.2d 1222, 1236 (3d Cir. 1986), *cert. denied* 479 U.S. 1031 (1987) (holding that the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea), *with* Computer Assoc. Int'l, v. Altai, Inc., 982 F.2d 693, 705-12 (2d Cir. 1992) (rejecting Whelan's general formulation that a program's overall purpose equates with the program's idea as descriptively inadequate and adopting a three step, abstraction/filtration/comparison test for deciding cases claiming non-literal copying).

The secondary literature on this issue is voluminous. The sources cited and relied on in the *Altai* case alone, for example, include the following: Menell, *supra* note 67, at 1051-57 (1989); Steven R. Englund, Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 867-73 (1990); Thomas M. Gage, Note, *Whelan Associates v. Jaslow Dental Laboratories: Copyright Protection for Computer Software Structure-What's the Purpose?*, 1987 WIS. L. REV. 859, 860-61 (1987); Marc T. Kretschmer, Note, *Copyright Protection For Software Architecture: Just Say No!*, 1988 COLUM. BUS. L. REV. 823, 824-27 (1988); Peter G. Spivack, Comment, *Does Form Follow Function? The Idea/Expression Dichotomy In Copyright Protection of Computer Software*, 35 U.C.L.A. L. REV. 723, 729-31 (1988). *See generally* 3 NIMMER, *supra* note 63, § 13.03[F].

74. *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930), *quoted in* *Computer Assoc.*, 982 F.2d at 701. *See generally* Nimmer, *supra* note 63, §13.03[A][1].

75. *See Computer Assoc.*, 982 F.2d at 703 (citing *Baker v. Selden*, 101 U.S. 99 (1879); *Mazer v. Stein*, 347 U.S. 201, 217 (1954)).

76. *Feist Publications, v. Rural Tel. Serv. Co.*, 111 S. Ct. 1282 (1992).

77. The court in *Sega* stated:

Computer programs pose unique problems for the application of the "idea/expression distinction" that determines the extent of copyright protection. To the extent that there are many possible ways of accomplishing a given task or fulfilling a particular market demand, the programmer's choice of program structure and design may be highly creative and idiosyncratic. However, computer programs are in essence utilitarian articles—articles that accomplish tasks. As such, 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. . . . Because of the hybrid nature of computer programs, there is no settled standard for what is protected expression and what is unprotected idea. . . .

1993 U.S. App. LEXIS 78, at *40-42.

78. The first generation of cases had a direct bearing on the permissibility of reverse engineering because, if a publicly distributed object code were left altogether unprotected by copyright, as some commentators contended it should be, see *supra* note 67, there would be no legal constraint on decompiling or disassembling it, though the results of decompilation or disassembly might be found to constitute an infringing copy or adaptation of the original source-code version of the program. As we shall see, *infra* text following note 230, the second generation of copyright cases may have an indirect bearing on the reverse engineering question.

79. See e.g., Menell, *supra* note 67.

80. See *supra* note 60 for the text of § 106, which states that the exclusive rights specified in § 106 are subject to §§ 107-120 on the Copyright Act.

81. See *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255 (5th Cir. 1988).

82. *Id.* at 261. For the text of § 117(1), see *infra* note 240.

83. *Id.*

84. 975 F.2d 832 (Fed. Cir. 1992).

85. 1993 U.S. App. LEXIS 78 (9th Cir. 1993).

86. *Atari*, 975 F.2d at 843-44; *Sega*, 1993 U.S. App. LEXIS 78, at *52-53.

87. *Atari*, 975 F.2d at 843-44; *Sega*, 1993 U.S. App. LEXIS 78, at *52-53.

88. 847 F.2d 255 (5th Cir. 1988). For criticisms of the *Vault* decision, see Part IV(D)(4).

89. *Id.* at 270.

90. *Atari*, 975 F.2d at 836; *Sega*, 1993 U.S. App. LEXIS 78, at *3. For *Sega's* and *Nintendo's* justifications for using a lockout device, see *infra* note 193.

91. *Atari Games Corp. v. Nintendo of Am.*, 18 U.S.P.Q. 1935, 1936 (1991) ("The security system consists of a patented computer chip located in both the [game console] and in authorized game packs."); *Atari*, 975 F.2d at 836 ("Atari could not break the IONES program code by monitoring the communication between the master and slave chips.").

92. *Atari*, 975 F.2d at 836.

93. *Atari*, 975 F.2d at 836. Chemically peeling the chip constituted reverse engineering of the semiconductor chip as permitted by section 906 of the Semiconductor Chip Protection Act, 17 U.S.C. § 906 (1988), but did not amount to reverse engineering of the computer program, which remained in object-code form. For a discussion of the difference between reverse engineering of semiconductor chips and computer programs, see *supra* note 57.

94. *Atari*, 975 F.2d at 836.

95. *Id.* at 837. Specifically, the attorney claimed that the copy was needed for litigation. The court in *Atari* ruled that Atari had no reasonable apprehension of litigation when it obtained the copy of *Nintendo's* program from the Copyright Office, and that Atari used the copy thus obtained to facilitate its reverse engineering of *Nintendo's* program. *Id.* at 836, 841. For the Copyright Office rules that

the attorney violated, see *supra* note 31.

96. *Id.* at 836. For a discussion of the Copyright Office regulations that were violated, see *infra* notes 204-206 and accompanying text.

97. *Id.* at 837. Although Atari's patent infringement claim was not discussed in the court of appeal's opinion, the existence of the patent claim gave the court of appeals for the Federal Circuit exclusive jurisdiction to review the trial court decision. *See* 28 U.S.C §§ 1292, 1295, 1338 (1988).

98. *Id.* at 835, 837.

99. *Id.* at 837.

100. *Id.* at 847.

101. *Id.* 843.

102. *Sega*, 1992 U.S. App. LEXIS 78, at *4-7.

103. *Id.*

104. *Id.* at *8.

105. *Id.*

106. *Id.* at *9.

107. *Id.* at *9-10. The file contained approximately twenty to twenty-five bytes of data. *Id.* at 9. The court noted that Accolade's games contain a total of 500,000 to 1,500,000 bytes. The court did not indicate how many bytes of data were contained in Sega's lockout or game programs. However, in a footnote added in its amended opinion, the court noted that Sega's "key" appears to be functional and is of such de minimis length that it was probably unprotected under the words and short phrases doctrine spelled out in the Copyright Office's regulations, 37 C.F.R. § 202.1(a); *Sega*, 1993 U.S. App. LEXIS 78, at *41 n.7.

108. *Sega*, 1993 U.S. App. LEXIS 78, at *10.

109. *Id.* at *11.

110. *Id.*

111. 785 F. Supp. at 1399-1400.

112. 1993 U.S. App. LEXIS 78, at *67-68. *See also* *Brief Amicus Curiae of Eleven Copyright Law Professors in Sega Enterprises Ltd. v. Accolade, Inc.*, 33 JURIMETRICS J. 147 (1992).

113. *See id.* at 256.

114. *Id.*

115. *Vault Corp. v. Quaid Software Ltd.*, 655 F. Supp. 750, 755, *aff'd* 847 F.2d 255, 270 (5th Cir. 1988).

116. *Id.*

117. *Id.*

118. EC Directive, *supra* note 3.

119. *Id.* See generally Thomas Dreier, *The Council Directive of 14 May 1991 on the Legal Protection of Computer Programs*, 9 EUR. INTEL. PROP. REP. 319 (1991); Vinje, *supra* note 15.

120. EC Directive, *supra* note 3. Articles 5 and 6 must be read in conjunction with Article 4, which enumerates the "restricted acts" (i. e., the exclusive rights) of the copyright holder of a computer program. The text of Article 4 is as follows:

Article 4

Restricted Acts

Subject to the provisions of Articles 5 and 6, the exclusive rights of the rightholder within the meaning of Article 2, shall include the right to do or to authorize: . . .

(a) the permanent or temporary reproduction of a computer program by any means and in any form, in part or in whole. Insofar as loading, displaying, running, transmission or storage of the computer program necessitate such reproduction, such acts shall be subject to authorization by the rightholder;

(b) the translation, adaptation, arrangement and any other alteration of a computer program and the reproduction of the results thereof, without prejudice to the rights of the person who alters the program;

(c) any form of distribution to the public, including the rental, of the original computer program or of copies thereof. The first sale in the Community of a copy of a program by the rightholder or with his consent shall exhaust the distribution right within the Community of that copy, with the exception of the right to control further rental of the program or a copy thereof.

121. The full text of section 107 is as follows:

Notwithstanding the provisions of sections 106 and 106A, the fair use of a copyrighted work, including such use by reproduction in copies or phonorecords or by any other means specified by that section, for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright. 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 amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
- (4) the effect of the use upon the potential market for or value of the copyrighted work.

The fact that a work is unpublished shall not itself bar a finding of fair use, if such finding is made upon consideration of all the above factors.

17 U.S.C. § 107 (1988 and supp. 1993).

122. 17 U.S.C. § 107 (1988).

123. 975 F.2d 832 (Fed. Cir. 1992).

124. 977 F.2d 1510 (9th Cir. 1992), *amended*, 1993 U.S. App. LEXIS 78.

125. 623 F. Supp. 1485 (D. Minn. 1985).

126. *Id.* at 1494 (verbatim copying), 1493-98 (substantial similarity).

127. *Id.* at 1501 n.17.
128. 10 U.S.P.Q.2d 1177 (N.D. Cal 1989).
129. *Id.* at 1184.
130. *Id.* at 1187.
131. *Atari*, 18 U.S.P.Q.2d at 1935.
132. *Id.* at 1939 (*quoting* E.F. Johnson, 623 F. Supp. at 1501).
133. *Id.*
134. *Sega*, 785 F. Supp. at 1396.
135. *Id.* at 1400.
136. *Id.* at 1397.
137. *Id.* at 1398.
138. Pub. L. No. 96-517, 94 Stat. 3028 (Dec. 12, 1980) (codified at 17 U.S.C. §§ 101, 117 (1988)).
139. This amendment added a definition of "computer program" to section 101 of the 1976 Act and revised section 117. In the original section 117 of the Copyright Act of 1976, Congress specifically declined to adopt a position on the copyrightability of computer programs, deferring a decision on the question until the previously created National Commission on New Uses of Technology (CONTU) issued a final report. Final Report of the National Commission on New Technological Uses of Copyrighted Works (July 31, 1978)[hereinafter CONTU Final Report]. The Computer Software Copyright Act of 1980 essentially adopted the recommendations of CONTU. *See generally* Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 DUKE L.J. 663, 694 (1984).
140. *See* STERN, *supra* note 41, § 1.4 (1986). *See also* Charles R. McManis, *International Protection for Semiconductor Chip Designs and the Standard of Judicial Review of Presidential Proclamations Issued Pursuant to the Semiconductor Chip Protection Act of 1984*, 22 GEO. WASH. J. INT'L L. & ECON. 31 (1988).
141. Semiconductor Chip Protection Act of 1984, 17 U.S.C. §§901 *et. seq.* (1988).
142. 17 U.S.C. § 906(a) (1988).
143. *See* H.R. Rep. No. 781, 98th Cong., 2d Sess., *reprinted in* 5 U.S.C.CA.N. 5750, 5771 (1984) [hereinafter Chip Act Legislative History]. For an assertion that a similar reverse engineering custom exists in the software industry, see *infra* note 163.
144. *Id.* For the distinction between reverse engineering of semiconductor chip designs and reverse engineering of computer programs, see *supra* note 57.
145. *Sega*, 785 F. Supp. at 1398.
146. Chip Act Legislative History, *supra* note 143, at 5758.
147. *Id.* at 5777.

148. *Id.* at 5759.

149. *Sega*, 785 F. Supp. at 1398.

150. *Sega*, 977 F.2d at 1521 (citing Chip Act Legislative History, *supra* note 143, at 5757-59).

151. Section 905 of the Chip Act states that the owner of a "mask work" (i.e., a chip design) has the exclusive rights to do and to authorize any of the following:

(1) to reproduce the mask work by optical, electronic, or any other means;

(2) to import or distribute a semiconductor chip product in which the mask work is embodied; and

(3) to induce or knowingly to cause another person to do any of the acts described in paragraphs (1) and (2).

17 U.S.C. § 905 (1988). To compare § 905 with the exclusive rights granted under § 106 of the 1976 Copyright Act, 17 U.S.C. § 106 (1988), see *supra* note 60.

152. See 17 U.S.C. § 912(a) (1988).

153. See 17 U.S.C. 117 (1988).

154. 17 U.S.C. §§ 108-20 (1988). These sections contain a number of "safe-haven" provisions spelling out clearly permissible uses that would otherwise violate §106. Among these is §117. If uses that do not fall within these safe haven provisions could not constitute fair use, section 107 would have no function.

155. *Atari*, 975 F.2d at 843.

156. WILLIAM F. PATRY, *THE FAIR USE PRIVILEGE IN COPYRIGHT LAW* 401 (1985).

157. See generally Committee Report, *supra* note 19.

158. *Id.* at 139.

159. See generally *LaST Frontier Conference Report on Copyright Protection of Computer Software, Conference Report*, 30 JURIMETRICS J. 15 (1989) [hereinafter Conference Report].

160. *Id.* at 24-25. A number of these same academics filed an amicus brief in *Sega*, arguing for the position the courts of appeals eventually adopted. See *Brief Amicus Curiae of Eleven Copyright Law Professors in Sega Enterprises Ltd. v. Accolade, Inc.*, 33 JURIMETRICS J. 147 (1992).

161. Conference Report, *supra* note 159, at 24-25.

162 Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977 (1993). For a critique of Professor Miller's arguments, see *infra* notes 192, 203, 214, 227-28, 230 and 238.

163. See, e.g., Clapes, *supra* note 67, at 1501:

The vitality of the software industry could be imperiled by a drastic limitation of the scope of copyright protection available to authors of computer programs. Yet, largely on the basis of misapprehensions and ignorance, movements to do exactly that are gathering adherents in this country and abroad.

Anthony L. Clapes is identified as Senior Corporate Counsel of the IBM Corporation. *But Cf.* Johnson-Laird, *supra* note 13, at 354:

Reverse engineering is practiced by all programmers, even those that work for those companies such as IBM, Apple, Ashton-Tate, and Microsoft, who wish to outlaw reverse engineering.

164. 17 U.S.C. § 107(1) (1988).

165. *Sega*, 785 F. Supp. at 1398.

166. Committee Report, *supra* note 19, at 139.

167. *Id.* (quoting *Sony Corp. of Am. v. Universal City Studios*, 464 U.S. 417, 449 (1984)).

168. *Id.*

169. *Id.* at 140.

170. *Id.*

171. For a more complete discussion of the "nature" of computer programs, see *infra* Part IV(C)(3)(b), discussing the second factor in the four-factor test.

172. 785 F. Supp. at 1398, (quoting PATRY, *supra* note 156, at 401).

173. 785 F. Supp. at 1398.

174. *Whelan Assoc. v. Jaslow Dental Lab.*, 797 F.2d 1222 (3d Cir. 1986), discussed at text accompanying *infra* note 232.

175. *Computer Assoc.*, 982 F.2d 693, also discussed at text accompanying *infra* note 231.

176. Committee Report, *supra* note 19, at 140.

177. *Sega*, 785 F. Supp. at 1398 (quoting PATRY, *supra* note 156 at 401).

178. *Sega*, 1993 U.S. App. LEXIS at 78, *38.

179. For such an inquiry, see *infra* notes 222-23, 229-30 and accompanying text.

180. *Atari*, 975 F.2d at 843.

181. *Sega*, 1993 U.S. App. LEXIS 78, at *33.

182. *Id.* at *34.

183. *Sega*, 1993 U.S. App. LEXIS 78.

184. *Sega*, 1993 U.S. App. LEXIS 78 at *36, (citing *Feist*, 111 S. Ct. 1282 (1991)). Although in *Sony Corp. of Am. v. Universal City Studios*, 464 U.S. 417, 449 (1984), duplication of the entire copyrighted work was for a non-commercial purpose, thus weighing more clearly in favor of a fair use finding, the use was also a "non-productive" use, in the sense that it did not lead to the production of additional works of authorship. While the Court in *Sony* rejected the argument that a non-productive use could not be considered a fair use, and later in *Harper & Row v. Nation Enterprises*, 471 U.S. 539, 555 (1985), rejected the argument that the particular productive use of an unpublished work was a fair use, it seemed to concede the relevance of the productivity of a use when it stated in that "a productive use is merely one factor in a fair use analysis." 471 U.S. 539, 555, 561 (1985).

185. Atari had alleged the existence in Nintendo's licensing agreements of a "tie-out" provision that conditioned the license of its lockout program on the licensee's agreement not to produce games for any other home video system for two years after the first sale of any Nintendo-compatible game. *Atari*, 975 F.2d at 832, 84. Accolade's misuse defense was based on unspecified tying allegations. *Sega*, 785 F. Supp. at 1399.

186. 975 F.2d at 846 (citing, inter alia, *United States v. Loew's, Inc.*, 371 U.S. 38 (1962); *Lasercomb Am. v. Reynolds*, 911 F.2d 970 (4th Cir. 1990)). *Loew's* was said to have given tacit approval to the copyright misuse defense, and *Lasercomb* specifically sustained such a defense. *Lasercomb* was the owner of a CAD/CAM computer program and had licensed some copies to Holiday Steel, a manufacturer of steel rule dies. *Lasercomb*, 911 F.2d at 971. The licensee, with the help of Reynolds, a computer programmer, created and sold a program which was almost entirely a copy of *Lasercomb's* licensed software. *Id.* *Lasercomb* sued the programmer (and his employees as well) on grounds of copyright infringement. *Id.* at 972. Defendants did not deny they had copied plaintiff's software, but alleged that *Lasercomb* had misused its copyright by including in its licensing agreement two clauses in which the licensee agrees not to compete with the licensor. *Id.* at 972-73. The provisions read as follows:

D. Licensee agrees during the term of this Agreement that it will not permit or suffer its directors, officers and employees, directly or indirectly, to write, develop, produce or sell computer assisted die making software. E. Licensee agrees during the term of this Agreement and for one (1) year after the termination of this Agreement, that it will not write, develop, produce or sell or assist others in the writing, developing, producing or selling computer assisted die making software, directly or indirectly without *Lasercomb's* prior written consent. Any such activity undertaken without *Lasercomb's* prior written consent shall nullify any warranties or agreements of *Lasercomb* set forth herein.

Id. at 973.

The term of the agreement was 99 years, possibly longer than the term of the copyright itself. *Id.* The court in *Lasercomb* upheld the defendant's copyright misuse defense, concluding that "a misuse of copyright defense is inherent in the law of copyright just as a misuse of patent defense is inherent in patent law," and that even though the copyright owner's conduct may not reach the level of antitrust violation, the clauses were still in conflict with the goals of copyright law. *Id.* at 973. In *Lasercomb*, the court was clearly more concerned with promoting the goals of copyright law than with finding an antitrust violation. Indeed, the anticompetitive effects of the agreement seem almost irrelevant to the court's finding of misuse. Holiday Steel was not a CAD/CAM software producer, therefore the agreement would not restrain existing competition. Furthermore, having no experience in producing and marketing software, Holiday did not appear as a threatening potential competitor either to *Lasercomb* or to the rest of the market. The tying-out clauses would not result in higher prices either for CAD/CAM programs or for steel dies. Likewise, reduction in the output of either was not to be expected. The anticompetitive effect of the clauses was a temporary barrier to the entry of an insignificant competitor.

For a summary of the defense of patent misuse, see 5 DONALD S. CHISUM, PATENTS § 19.04 (1993) and ERNEST B. LIPSCOMB III, WALKER ON PATENTS, § 28.32-.36 (1984, Cum. Supp. 1992). The concept of patent misuse involves the interplay of two elements: the inherent rights to the invention covered by the patent (basically the rights to exclude others from making, using, and selling the invention) and the requirement that competition beyond the scope of the patent rights not be hindered by the patentee seeking artificially to increase its revenue. The test to detect misuse, therefore, includes two sequential steps. First, courts must assess whether the patentee is actually enforcing (or otherwise exercising) its rights. The patentee may be entitled to engage in conduct which, in the absence of a patent, would constitute an antitrust violation -- e.g., fixing prices at which a licensee will sell the licensed product. When the patentee acts beyond the scope of the patent -- e.g., attempts to enforce an invalid patent -- there is technically no misuse as such, though some courts characterize it as such, because no existing patent right is being invoked. Second, when the patentee is acting within the scope of the patent, the court must assess whether the patent's practices have nevertheless impaired competition outside the scope of its patent rights (even though perhaps not to the point of violating the antitrust laws) and thus tampered with the patent's capacity to be used as a meter to measure accurately the invention's economic value. This theory, in a less elaborate manner, has been recently stated by the Court of Appeals for the Federal Circuit, in *Mallinckrodt, Inc. v. Medipark, Inc.*, 976 F.2d 700, 703 (Fed. Cir. 1992) ("The concept of patent misuse arose to restrain practices that did not in themselves violate any law, but that drew anticompetitive strength from the patent right, and thus were deemed to be contrary to public policy."). For an application of this test, see *infra* note 191.

187. *Atari*, 975 F.2d at 846.

188. *Id.* Even had Atari's hands been clean, its tie-out allegation, see *supra* note 185, would have been unsuccessful if the restrictive clause merely involved the licensed rights. Although as a matter of law, licensors may not prohibit licensees from dealing with competing goods, they may restrain the disposition by licensees of the products embodying the licensed software. *Cf.* *Bela Seating Co. v. Poloron Prod.*, 438 F.2d 733 (7th Cir. 1971, cert. denied, 403 U.S. 922 (1971)). In that case, *Bela Seating* and two patent licensees

had entered into cross-licensing agreements which included the clause that they would manufacture "table arm chairs embodying the invention . . . but not of substantially identical design." *Id.* at 738-39. The court held that such a restriction was not misuse because "[a] licensor may restrict the licensee's manufacture of the patented item to a specific use or design. . . . [T]he restrictions in the Bela agreements are limited to the manufacture of chairs covered by the claims of the patent in suit." *Id.* at 739. Likewise, Nintendo's restrictive clause could be construed as covering only video games which embodied its copyrighted software, i.e., the lockout device. However, it is clear that Nintendo's purpose was much broader. In order to become a Nintendo licensee, Atari would be prohibited from supplying games of its own creation for other home video systems for two years. *See supra* note 185. Tie-outs are subject to the same rule of reason approach and to the same required findings of actual market power as a tie-in. *See infra* note 190. If market power were found, however, it appears that it might be very difficult for Nintendo to show a good business justification for the tie-out provision, since the clause would prohibit Atari from supplying games for other home video systems--which did not require the use (hence, the license) of the lockout program. *See generally* Donald A. Gregory & James J. Trussell, *Tie-Outs -- Misuse of U.S. Patents*, 14 EUR. INTEL. PROP. REP. 317 (1992).

189. *Sega*, 785 F. Supp. at 1399.

190. *Id.* The basis for Accolade's tying allegation is not clear. Apparently, Accolade was attempting to claim that by using the lockout device, Sega intended to tie the sale of Sega-compatible cartridges (the tied product) to the sale of Sega's consoles (the tying product). Sega required those, such as Accolade, who wished to create Sega-compatible games, to agree to make Sega the exclusive manufacturer of all games that they produced. *Sega*, 1993 U.S. App. LEXIS 78, at *4. The problem with the tying theory, however, is that Sega does not have monopoly power in the game console market. Nintendo, after all, offers a competing console. Tying is a specific antitrust violation with specific elements. Clayton Act § 3, 15 U.S.C. §14 (1991). First, it requires the tying of two separate products. Second, competition must be substantially lessened or there must be a tendency to create a monopoly with respect to the tied product. Third, the seller must have market power over the tying product. *See* PHILLIP E. AREEDA, 9 ANTITRUST LAW § 1702 (1991) (stating five conditions for an illegal tie-in: (1) two products, (2) a tie between them, (3) substantial economic power over tying product by suppliers, (4) anticompetitive effort over (5) a not insubstantial volume of commerce. Apparently, Sega did wish to remain the sole manufacturer of Sega-compatible cartridges. As Sega's customers would be locked-in to Sega consoles, Sega could then dominate cartridge market. The first and second elements thus might be present. Sega's requirement that those, such as Accolade, who wished to create Sega-compatible games, must agree to make Sega the exclusive manufacturer of all games they produced, moreover, could restrain interbrand competition in the cartridge market. *Sega*, 1993 U.S. App. LEXIS 78, at *4. Nevertheless, copyright (by analogy with the current understanding of patents) does not automatically give rise to monopoly power over the copyrighted products (i.e. the lockout device in the console and cartridge). *Cf.* 35 U.S.C. § 271(d)(5) (added by Pub. L. No. 100-703 § 201, 102 Stat. 4676, (1988), stating that no patent owner shall be denied relief or deemed guilty of misuse by reason of having conditioned the license of any rights to the patent or the sale of the patented product on the acquisition of a license to rights in another patent or purchase of a separate product unless in view of the circumstances that patent owner has market power in the relevant market for the patent or patented product on which the license or sale is conditioned). Thus, the defendant in a copyright misuse case must give evidence of the copyright owner's market power over the tying product. Most probably, Accolade's allegation was rejected because this evidence was not supplied.

Under the so-called 'Jerrold rule,' *United States v. Jerrold Elec. Corp.*, 187 F. Supp. 545, 555-58 (E.D. Pa. 1960), *reh'g denied*, 365 U.S. 890 (1961), tying may be permissible if there is a sufficient business justification. This rule of reason approach was brought into the Patent Act by the 1988 Amendment, 35 U.S.C. § 271(d)(5), which requires courts to take a "view of the circumstances" of the license. *Sega* might have an acceptable justification because it could well allege that the exclusive dealership was aimed at keeping uniform quality standards in the manufacture of game cartridges as well as the rendering of technical assistance to consumers.

However, in both *Atari* and *Sega*, the defendants failed to raise the only antitrust argument that plausibly arises from the facts--namely, a Sherman Act § 2 violation--an actual or attempted monopolization of the cartridge market. 15 U.S.C.A. §2 (1991). In effect, inasmuch as the video game systems accept only compatible cartridges, the relevant market in both cases would be strictly limited to compatible cartridges. A cartridge for the "Genesis" console does not compete with a cartridge for a "Nintendo" console, as consumers cannot shift from one system to another, unless they also buy the other basic console. Therefore, Sega and Nintendo had the power to raise prices of cartridges provided they do not reach a level at which buyers would be induced to shift the console they use. Once the relevant market is defined in this manner, it follows that Sega and Nintendo, by concealing the codes and invoking copyright protection, were engaged in practices aimed at barring the entry of competitors into the cartridge market. *See Digidyne Corp. v. Data Gen. Corp.*, 734 F.2d 1336 (9th Cir. 1984).

Digidyne is a typical tying case. Data General manufactured a computer system, consisting of a CPU and a copyrighted operating system, called RDOS. *Id.* at 1338. Digidyne produced CPUs that could only operate with RDOS because they emulated Data General's CPUs. *Id.* Data General refused to license RDOS for anyone who did not buy Data General's CPU. *Id.* In assessing Data General's market power, the court followed the now overruled theory that copyrights, by definition, afford monopoly power. *Id.* at 1341-42. But,

equally importantly, the court acknowledged that "many of defendant's customers were *locked in* to the use of RDOS." *Id.* at 1342. Data General sold CPUs to equipment manufacturers who created a complete computer system for resale. These manufacturers designed application software, which could not run without Data General's operating system, RDOS. *Id.* The conversion of the application software so that it could operate with a system other than RDOS was out of the question as it was not economically viable. From these facts the court concluded: "Defendant's operating system has been shown to be unique as a matter of law and distinctively attractive as a matter of fact. Defendant's initial leverage is magnified by the lock-in." *Id.* at 1342, 1343. Even though it is no longer good law to equate an intellectual property right with market power for purposes of establishing a Sherman Act § 1 tying violation, *Digidyne* remains a landmark case to the extent that it has acknowledged that the "locking-in" of customers may dramatically circumscribe the relevant market for a product to the product itself, for purposes of determining whether a Sherman Act § 2 violation has occurred.

Likewise, in both *Sega* and *Atari*, there is a product (the consoles) "locking in" buyers and a market over which the copyright owners intended to acquire monopoly power (the cartridge market). Thus, *Sega* and Nintendo might be deemed to have engaged in practices (namely, the enforcement of copyrights) aiming at maintaining monopoly power over the compatible-cartridge market. Even in the event monopoly power was not found, however, *Sega* and Nintendo nevertheless might be deemed misusers, as both were admittedly acting within the scope of their copyright (although in *Sega*, the court of appeals was not sure of that, see *infra* note 223) in a manner that, through the barrier their lockout devices created to the entry of competitors into the cartridge market, allowed them to raise prices.

191. *Sega*, 1993 U.S. App. LEXIS 78, at *51. The court said:

[T]he fact that computer programs are distributed for public use in object code form often precludes public access to the ideas and functional concepts contained in those programs, and thus confers on the copyright owner a de facto monopoly over those ideas and functional concepts. That result defeats the fundamental purpose of the Copyright Act. . . .

Id. Therefore, the court found in *Sega*'s use of its copyright the two elements that, combined, indicate the presence of misuse: a conflict with the statutory goals and an anticompetitive effect. The most controversial aspect of the misuse doctrine does not reside in its definition but, rather, in the fact that it can be raised as an affirmative defense even though the defendant has not been damaged by the misusing practice. See, e.g., *Morton Salt Co. v. C.S. Suppiger Co.*, 314 U.S. 488 (1942). In *Atari* and *Sega*, however, the defendants were both being adversely affected by the lockout devices.

192 Professor Miller, *supra* note 162, at 1019-20, claims that there is "no principled basis for the Ninth Circuit's definition of the relevant market in which monopoly control was to be prevented as the market for *Sega* cartridges. . . ." But see the discussion, *supra* note 190, of *Digidyne Corp. v. Data Gen. Corp.*, 734 F.2d 1336 (9th Cir. 1984), which provides just such a principled basis for the definition of the relevant market.

193 For a more extended discussion of this point, see *supra* note 190. Thereason *Sega* gave for developing the lockout device was concern over the rise of software and hardware piracy in Asia. Counterfeiters had discovered how to blank out the screen display of its trademark before selling pirated games as their own. 1993 U.S. App. LEXIS 78 at *7. The lockout device also gives the console producer the ability to exercise quality control over games that can be played on the console. However, as long as there is no confusion about the source of the game cartridge, its quality is of no legitimate concern to the console producer unless the cartridge could cause damage to the console.

194. *Sega*, 1993 U.S. App. LEXIS 78, at *38.

195 *Id.* at *53-55.

196 *Id.* at *8.

197 *Id.* at *10.

198. *Id.* at *59.

199. Committee Report, *supra* note 19, at 140. For a discussion of the Committee's failure to recognize that this characteristic of computer programs also plays a role in determining the purpose of the reverse engineering, see *supra* text accompanying note 171.

200. *Id.*

201. *See supra* note 171 and accompanying text.

202. *Sega*, 1993 U.S. App. LEXIS 78, at *44.

203. *Atari*, 975 F.2d at 843. Professor Miller protests that "[a]lthough it is true that the scope of protection given to different types of literary works may vary, they are all treated alike-absent statutory direction to do otherwise-under identical copyright principles, and the exclusive right to make intermediate copies is accorded to all copyright holders, as *Sega* itself recognized." Miller, *supra* note 163, at 1022 (citing *Sega*, 977 F.2d 1510, 1518-19). However, courts are directed by the language of § 107(2) to consider the "nature" of the copyrighted work, which presumably means not only those aspects of its nature that it has in common with all other types of copyrighted works, but also those aspects of its nature that are unique.

204. *Id.* at 836. The court in *Atari* ruled that Atari had no reasonable apprehension of litigation when it obtained the copy of Nintendo's program from the Copyright Office, and that Atari used the copy thus obtained to facilitate its reverse engineering of Nintendo's program. *Id.* It was this use of the unauthorized copy of Nintendo's program that the court found to constitute copyright infringement.

205. *See supra* notes 31-33 for a discussion of the regulations contained in C.F.R. § 202.20(c)(2)(vi)(vii) (1992). The Register of Copyrights is specifically authorized by section 408 of the 1976 Copyright Act to require or permit, for particular classes of works, the deposit of identifying material instead of copies or phonorecords. Pursuant to this authority, the Register adopted regulations in 1979 which permit the deposit of identifying material (in lieu of the statutory deposit) for secure tests and computer programs. *See generally* 3 MILGRIM, *supra* note 19, App. B18-5. These rules were subsequently amended a number of times. *See, e.g.*, 54 Fed. Reg. 13,173 (1989) (for notice of the Copyright Office's most recent amendment of its rule); *see also* 51 Fed. Reg. 34667 (1986) (where the Copyright Office summarized the results of its original rulemaking proceeding). The Copyright Office set forth the policy basis for the rule as follows:

On May 23, 1983, the Copyright Office published a Notice of Inquiry in the Federal Register requesting public comments on the deposit of material containing trade secrets. (48 FR 22951) . . .

The Copyright Office received a total of 41 responses from the notice of inquiry. The vast majority of the responses were from members of the computer industry and the overwhelming sentiment was in favor of establishing special deposit procedures to mitigate the alleged uncertainties associated with depositing in a public office, material containing trade secrets.

A number of comments addressed public policy issues concerning the establishment of special deposit provisions. Several of the comments expressed the view that trade secret protection and copyright advance similar societal goals, and therefore it is completely consistent to modify the deposit requirement in a way that would preserve trade secret protection fully. The Association of American Publishers argued that the deposit requirement was not intended to delineate the scope of a copyright claim through public disclosure, citing the Register's authority to determine the nature of deposited material under section 408(c)(1) and *National Conference of Bar Examiners v. Multistate Legal Studies*, 692 F.2d 478 (7th Cir. 1982), concerning deposit of secure tests. Only two comments argued in favor of a deposit that fully discloses and copyrightable content of registered material. One asserted that public disclosure through deposit was intended as a trade-off for receiving copyright protection, and the other argued that owners of intellectual property should elect either copyright protection or trade secret protection.

On the basis of the comments received, the Copyright Office has concluded that a case has not been made for establishment of a broad deposit exemption covering all material which could conceivably contain trade secrets. Of the submitted comments, only one came from outside of the computer industry. That comment came from a manufacturer of spare parts who argued that public inspection should be restricted on deposits of technical drawings and specifications.

On narrower grounds, however, the Copyright Office finds that particular problems of the computer industry do merit special attention. Many in the computer industry are concerned that the availability of registered computer programs for public inspection in the Copyright Office gravely jeopardizes trade secret protection. While no court has directly addressed the issue, it is clear that computer programs are valuable intellectual property whose owners are rightfully concerned about adequate protection for their works . . .

Frequently mentioned among the submitted comments was the proposal that the Copyright Office merely restrict public access to deposits of computer programs and other material containing trade secrets. The Copyright Office has concluded that such an approach would clearly violate section 705(b) mandating public inspection of deposits retained by the Copyright Office. As a result, this proposal has not been adopted.

Id.

In *National Conference of Bar Examiners v. Multistate Legal Studies*, 692 F.2d 478 (7th Cir. 1982), the court explicitly held that the Register of Copyrights has authority to promulgate the regulations requiring the Copyright Office to return the deposit of a secure test, such as the Multistate Bar Examination, and retain only such portions, description, or the like so as to constitute a sufficient archival record of the deposit. *Id.* at 486-87. The regulation thus promulgated was found neither to be inconsistent with the requirement in section 704(d) of the Copyright Act that the Register maintain the "entire deposit" of unpublished works during the term of copyright, nor to defeat the purpose of the registration provision, nor to undermine the constitutional requirement that federal copyright law "[p]romote the progress of Science and useful Arts." *Id.* In so holding, the court concluded that the statutory scheme of the Copyright Act, when viewed as a whole, negates the notion that the deposit provisions are for the purpose of public disclosure. *Id.*

Even granting all of the foregoing, however, one may question whether the Register of Copyrights has authority to promulgate regulations that, in effect, affirmatively provide federal trade secret protection for the source-code version of computer programs that are being publicly distributed in object-code form. As we will see in Part V, these regulations, like state law efforts to enforce contractual restrictions on reverse engineering of publicly distributed programs, may conflict with the preemptive policies of federal copyright and, particularly, federal patent law. The dual nature of computer programs makes them distinguishable from the secure tests involved in the *National Conference of Bar Examiners* case. The Copyright Office regulations, moreover, make it more difficult, rather than easier, for the public to ascertain the uncopyrightable ideas contained in the publicly distributed versions of computer programs.

206 *See supra* note 205.

207 101 U.S. 99, 1104 (1879)

208. *Id.* at 99.

209. 471 U.S. 539, 555 (1985).

210 The scope of fair use is generally narrower with respect to unpublished works. *Id.* at 551-54.

211. *Id.* at 563 (citing Gorman, *Fact or Fancy? The Implications for Copyright*, 29 J. COPYRIGHT SOC'Y 560, 561 (1982)).

212. *Feist*, 111 S. Ct. 1282, 1289 (1992).

213 *Sega*, 1993 U.S. App. LEXIS 78, at *40.

214. *Id.* Professor Miller loses sight of this critical aspect of a computer program's nature when he protests that the effect of the *Sega* decision is that popular computer systems will "lose" their copyright protection "merely because their popularity denominates them 'standards,' a proposition that would not be taken seriously if the copyrighted works were Steinbeck's *Grapes of Wrath*, Hemingway's *The Sun Also Rises*, or Miller's *Death of a Salesman*." Miller, *supra* note 162, at 1020. A common error of those advocating broad protection for computer programs and a ban on decompilation and disassembly is that they analogize the scope of copyright protection for computer programs (utilitarian works) to the scope of protection accorded to fictional (i.e., not utilitarian) literary works.

215. *Harper & Row*, 471 U.S. at 551-54. Section 107 of the Copyright Act was amended in 1992 to make explicit that unpublished works were subject to fair use. Act of Oct. 24, 1992, Pub. L. No. 102-492, 106 Stat. 3145 (1992). For the amended text of § 107, see note 121. The legislative history of the amendment indicates that it was not intended to affect the treatment of decompilation under the Copyright Act. *See* 137 CONG. REC. S.5648 (May 9, 1991).

216. *Sega*, 1993 U.S. App. LEXIS at 78, *48 n.9 (citing *Lewis Galoob Toys v. Nintendo of Am.*, 964 F.2d 965 (9th Cir. 1992)).

217 Committee Report, *supra* note 19, at 140.

218. *Id.*

219 *Id.* (citing *Sony Corp. of Am. v. Universal City Studios*, 464 U.S. 417, 449 (1984)).

220. *Id.*

221 The Court specifically recognized that the duplication in *Sony* was a "non-productive" use, in the sense that it did not lead to the production of additional works of authorship. In *Harper & Row*, however, the court stated that "a productive use is merely one factor in a fair use analysis." *Harper & Row*, 471 U.S. at 561 (1985).

222. 975 F.2d at 843.

223. The *Sega* amended opinion added a footnote rejecting *Sega's* belated suggestion that *Accolade's* use of the code which "unlocks" the Genesis III console is an unfair use of copyrighted expression. Said the court: "*Sega's* key appears to be functional. It consists merely of 20 bytes of initialization code plus the letters S-E-G-A. There is no showing that there is a multitude of different ways to unlock the Genesis III console." In addition, the court noted, "*Sega's* security code is of such *de minimis* length that it is probably unprotected under the words and short phrases doctrine." 1993 U.S. App. LEXIS 78, at *47 n.7. In other words, even considered separately and its functionality put aside, *Sega's* security code might be entirely quoted, regardless of the fair use defense.

The court distinguished the *Atari* case as having found that Nintendo's 10NES program generated an arbitrary data stream "Key" that unlocked the Nintendo console. *Id.* As the court of appeals in *Atari* made clear, the data stream did indeed do more than unlock the Nintendo console, and *Atari's* "Rabbit" program created by virtue of *Atari's* illicit reverse engineering of the Nintendo lockout program, generated an identical data stream, and thus borrowed more protectible expression than was necessary to perform the unlocking function. *Atari*, 975 F.2d at 839. What the court of appeals in *Atari* failed to address, however, was *Atari's* argument that, in copying what it did, *Atari* was simply planning ahead for the possibility that future Nintendo machines might be programmed to reject cartridges lacking these portions of the 10NES program and that it was entitled to copy more extensively from a copyrighted work so as to preempt future efforts by Nintendo to recognize and reject its unauthorized cartridges.

224. Committee Report, *supra* note 19 at 141, (quoting *Harper & Row v. Nation Enterprises*, 471 U.S. 539, 566 (1985)).

225 *Id.*

226. *Id.* at 141 (quoting S. Rep. No. 94-473, 94th Cong., 1st Sess. 65 (1975)).

227. See *Johnson-Laird*, *supra* note 13, at 343. The commentator points out that there is a great difference between the act of disassembling or decompiling and the act of creating a new program. *Id.* at 348. Moreover, contrary to what is generally believed (see, e.g., *Miller*, *supra* note 162, at 1026, who describes decompilation as a "relatively simple procedure") it is not that simple to generate a new, competitive program upon the information obtained from the disassembling. "[I]t takes more time to reverse engineer an entire program than to design a clone from scratch," the commentator adds. *Id.* at 348.

228. See *supra* notes 11-12 and accompanying text. Professor *Miller*, *supra* note 162, at 1026, sees the risk that trade secrets embedded in computer programs would be impaired in value as a reason not to recognize decompilation as a fair use. However, federal copyright law provides only a narrow form of trade secret protection for *unpublished* works. For published works, copyright protection does not and cannot protect trade secrets embedded in the work, but only protects expression. Likewise, state trade secret law provides no protection against reverse engineering of publicly distributed works. Finally, federal patent protection is available for patentable utilitarian features of computer programs, and specifies that all other utilitarian features of publicly distributed articles are to remain unprotected. See *infra* Part V(D).

229. In applying the four-factor test of section 107, the Conference Report, *supra* note 159, came to much the same conclusion as does this article, particularly with respect to the fourth factor. The report points out that the aim of § 107(4) is to assess the effect that the use of the copies can have on the market of the copyrighted work. *Id.* at 24-25. Therefore, the statute is not concerned with the effect that an otherwise noninfringing work (the new program, albeit a competing one) derived from the reverse engineered copy can have on the market of the original computer program.

230. *Hoehling v. Universal City Studios*, 618 F.2d 972, 980 (2d Cir. 1980). Professor *Miller* appears to make the converse mistake

when he states that one reason for prohibiting decompilation and disassembly is "[b]ecause the traces of copying can be disguised." Miller, *supra* note 162 at 1027. But "disguised copying" may not be impermissible copying of expression at all, but merely the use of the ideas expressed. The fairness of the initial copying (i.e., decompilation or disassembly), and the fairness of the subsequent use of the product of the reverse engineering, must each be determined on its own merits.

The larger difficulty with the argument of Professor Miller, *Id.* at 1013-32, and others who advocate broad copyright protection for computer programs with little, if any, decompilation or disassembly permitted, is that they deal with the scope of computer program copyright protection in isolation from other forms of legal protection. Because utilitarian features of computer programs are potentially patentable, federal patent law arguably takes primacy over copyright law in protecting patentable utilitarian features of a computer program and in declaring that all other utilitarian features of publicly distributed articles are to remain unprotected and subject to reverse engineering. *See infra* Part V(B)(2). If a user cannot get access to the utilitarian features of a computer program in order to incorporate those features into new computer programs, the latter policy of federal patent law is defeated.

The core of Professor Miller's argument is that, if decompilation and disassembly are permitted, and the "creativity and success by those in the position of Sega are to be penalized, not rewarded, later comers to the console market-or, indeed, any creative individual who might be unfortunate enough to create a `standard' product or feature-will be discouraged from providing that public benefit initially." Miller, *supra* note 162, at 1020. Contrary to Professor Miller's apparent assumption, there are at least two other types of incentives available to reward later comers-one provided by federal patent law for sufficiently innovative products or features, and the other provided by the very competitive process fostered by the federal antitrust laws that enabled Nintendo and Sega to enjoy such widespread commercial success.

231. 982 F.2d 693 (2d Cir. 1992).

232. 797 F.2d 1222 (3d Cir. 1986).

233 *Id.*

234. For a discussion of copyright and trademark misuse, see *supra* notes 185-98 and accompanying text. *See generally* Johnson-Laird, *supra* note 13. The commentator states that reverse engineering others' software programs is a standard of the industry even among those software producers which have strongly supported a ban on decompiling and disassembling source codes. *Id.* at 354.

235. EC Directive, *supra* note 3 (emphasis added).

236. Thomas Dreier, *The Council Directive of 14 May 1991 on the Legal Protection of Computer Programs*, 9 EUR. INTELL. PROP. REV. 319 (1991). This commentator quotes from the Communication of the Council's Common Position to the European Parliament:

Decompilation is permitted by Article 6 to the extent necessary to ensure the interoperability of an independently created computer program. Such a program may connect to the program subject to decompilation. Alternatively, it may compete with the decompiled program and in such circumstances will not necessarily connect to it. SEC 87 final-SYN 183, 18 January 1991, paragraph 4.7.

Id. at 315 n.57. *But see* Johnson-Laird, *supra* note 13, at 331 (contending that the EC Directive outlawed reverse engineering for the purposes of producing a competitive program).

237 Vinje, *supra* note 15, at 5: "Rather than permitting decompilation only for puposes of achieving the interoperability of an independently created program `with the original program,' the provision adopted. . . permitted the achievement of interoperability `with other programs'" (emphasis in original).

238. In *Sega*, there is some language suggesting by way of dictum the court's approval of reverse engineering aimed at developing a competitive product: "In any event, an attempt to monopolize the market by making it impossible for others *to compete* runs counter to the statutory purpose of promoting creative expression. . . ." *Sega*, 1993 U.S. App. LEXIS 78, at *53-55 (emphasis added). *But see* Miller, *supra* note 162, at 1015, who claims that "[a]s articulated in *Sega*, fair use permits decompilation only by someone who merely seeks to create a product that is complementary to, and not a substitute for, an existing work. . . ."

239 *Cf. supra* note 238, where Professor Miller claims that *Sega* would only permit decompilation for the purpose of creating a complementary work.

240. The complete text of section 117 is as follows:

Notwithstanding the provisions of section 106, it is not an 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.

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.

17 U.S.C. § 117 (1988).

241. 2 NIMMER, *supra* note 63, § 8.08 n.9.

242. *Id.*

243. CONTU Final Report, *supra* note 139, at 31-32 (referring to "creators, lessors, licensors and vendors of copies of programs" and comparing the section 117 rights of "rightful possessors" with the section 109(a) rights of owners of authorized copies of a copyrighted work.).

244. H.R. Rep. No. 1307, 96th Cong., 2d Sess., pt. 1, at 23 (1980), *reprinted in* 1980 U.S.C.C.A.N. 6460, 6482.

245. *See, e.g.*, Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 261 (5th Cir. 1988).

246. CONTU Final Report, *supra* note 139, at 31.

247. *Id.* at 31.

248. *Id.*

249. *Id.*

250. *Id.* at 32.

251. *Id.*

252. *Id.*

253. *Id.* at 31.

254. *Id.* at 33.

255. *Id.*

256. *See, e.g.*, *infra* notes 284-85 and accompanying text.

257. CONTU Report, *supra* note 139, at 31-32.

258. *See id*

259 Among the terms most commonly included in these so-called "shrink-wrap," "box-top" or "tear-open" licenses or contracts are: 1) clauses specifying that the software proprietor retains title to the software or diskette and that the user is merely a bailee or lessee rather than the owner; 2) clauses prohibiting the rental of the software; 3) clauses limiting the use of the computer program to a single computer; 4) clauses disclaiming implied warranties or merchantability or fitness for a particular purpose; and 5) clauses prohibiting any modification or disassembly of the computer program or its incorporation into any other program. *See generally* BENDER, *supra* note 1, § 4A.02[4]; Richard Stern, *Shrink-wrap Licenses of Mass Marketed Software: Enforceable Contracts or Whistling in the Dark?* 11 RUTGERS COMPUTER & TECH. L.J. 82 (1985).

At least one of the concerns that shrink-wrap licenses have attempted to address has recently been the subject of federal legislation. The Computer Software Rental Amendments Act of 1990, Pub. L. No. 101-650, 104 Stat. 5128 (1990) amends section 109 of the 1976 Copyright Act so as to prohibit, until expiration of the amendment on October 1, 1997, the unauthorized rental, lease or lending of software for a direct or indirect commercial purpose. This amendment in effect limits the application of the "first sale" doctrine embodied in section 109, which ordinarily limits the exclusive distribution right of a copyright owner to the first sale of a copyrighted work, while permitting the owner of an authorized copy to sell or otherwise dispose of that particular copy. The rationale for eliminating software rentals as a permissible method of disposing of a lawfully obtained copy of a copyrighted work is that such rentals, like phonorecord rentals, which section 109 also excepts from the first sale doctrine, are primarily designed to permit unauthorized copying of the rented work. H.R. Rep. 735, 101st Cong., 2d Sess. 7-10 (1990), *reprinted in* 1990 U.S. C.C.A.N. 6935, 6938-41.

260. *See* 2 BENDER, *supra*, note 1, § 4A.02[4]; Richard Stern, *supra* note 259.

261 *Id.*

262. *See infra* notes 279-80-and accompanying text, andnote 313..

263. LA. REV. STAT. ANN. § 51:1961-66 (West 1987); Ill. Pub. Act 84-901, eff. July 1, 1986.

264 LA. REV. STAT. ANN. § 51:1961-66 (West 1987).

265. 847 F.2d 255 (5th Cir. 1988).

266 *Id.* at 270.

267. Vault, 847 F.2d 255 (5th Cir. 1988). For a description of the reverse engineering employed by Quaid, see notes 113-17 and accompanying text.

268. *Id.* at 264.

269. *Id.*

270. *Id.* at 256-57.

271. *Id.* at 261. For the text of section 117(1), see *supra* note 240.

272. Vault, 847 F.2d. at 261.

273 *Id.*

274. 847 F.2d at 267, (citing Litchfield v. Spielberg, 736 F.2d 1352, 1357 (9th Cir. 1984), *cert. denied*, 470 U.S. 1052 (1985)).

275. 847 F.2d at 268 (distinguishing Whelan Assoc. v. Jaslow Lab., 797 F.2d 1222, 1244 (3d Cir. 1986) and Midway Mfg. Co. v. Artic Int'l, 704 F.2d 1009 (7th Cir. 1983), *cert. denied*, 464 U.S. 823).

276. 847 F.2d at 268 (distinguishing *Midway*, 704 F.2d at 1013-14).
277. *Sony*, 464 U.S. 417, 447-55 (1984).
278. 847 F.2d at 264.
279. *Vault*, 847 F.2d at 270, (citing *Sears, Roebuck & Co. v. Stiffel*, 376 U.S. 225, 229 (1964)("[w]hen state law touches upon the area of [patent or copyright statutes], it is 'familiar doctrine' that the federal policy 'may not be set at naught, or its benefits denied' by the state law.")).
- 280 *Id.*
281. John M. Conley & Vance.T. Brown, *Revisiting § 117 of the Copyright Act: An Economic Approach*, COMPUTER LAW., Nov. 1990, at 1, 9.
282. *Id.* at 5.
283. *Id.*
284. *Id.* (citing CONTU Final Report, *supra* note 139, at 14). *But see supra* text accompanying notes 256-58.
285. *See Foresight Resources Corp. v. Pfortmiller*, 719 F. Supp. 1006 (D. Kan. 1989). The court in *Pfortmiller* merely stated that a licensing agreement accompanied the plaintiff's program, without specifying whether it was a shrink wrap license or not. *Id.* at 1010. For further discussion of copyright and patent preemption of shrink wrap licenses, see *infra* Part V.
286. Conley & Brown, *supra* note 281, at 5, 9.
287. *Id.*
288. CONTU Final Report, *supra* note 139, at 31.
289. Conley & Brown, *supra* note 281, at 5-6.
- 290 See text following note 269, *supra*.
291. *Micro-Sparc, Inc. v. Amtype Corp.*, 592 F. Supp. 33, 35-36 (D. Mass. 1984); *Atari, Inc. v. JS&A Group*, 597 F. Supp. 5, 9 (N.D. Ill. 1983).
292. 847 F.2d at 266-67.
293. *See, e.g.*, section 108(c) of the Copyright Act of 1976, which gives certain libraries and archives a right to reproduce a copy of a published work "solely for the purpose of replacement of a copy . . . that is damaged, deteriorating, lost, or stolen, if the library or archives has, after a reasonable effort, determined that an unused replacement cannot be obtained at a fair price." 17 U.S.C. § 108(c) (1988).
- 294 *Sega*, 1993 U.S. App. LEXIS 78, at *28. The *Sega* court concluded that Accolade's reverse engineering of Sega's object code went far beyond the uses contemplated by CONTU and authorized by section 117. *Id.* The court thus found it unnecessary to decide the exact scope of section 117. *Id.* at *26 n.6.
295. *Id.* at *26.
296. *Vault*, 847 F.2d at 261.

297. See text following note 269, *supra*, and text accompanying note 291, *supra*.

298 See text accompanying *supra* note 293

299 A connection may well exist between the general use of "CopyWrite" by pirates and the loss of "PROLOK's" economic value. Although VCR's made by Sony did not destroy the economic value of broadcast movies, see *supra* notes 217-21 and accompanying text, "CopyWrite" and similar programs may be the determinant cause for the discontinuance in the use of protected diskettes by software producers.

300 *But see infra* note 341-43 and accompanying text, suggesting that § 117 could be interpreted to permit reverse analysis as a per se fair use, without any need to consider the § 107 four-factor test.

Professor Miller, *supra* note 162, argues that, to the contrary, the "circumscribed freedom that section 117 gives the program acquirer to make a copy or adaptation for internal use certainly carries with it the negative implication that the statute's permission does not extend to commercial competitors, whose copying and adaptation is not 'an essential step' in their own 'utilization' of the program." Miller, *supra* note 162, at 1023. This argument, however, fails to recognize that § 117 is merely a "safe-haven" provision that carries no negative implication at all for the interpretation of § 107. If uses not falling within § 117, or any of the other safe-haven provisions contained in §§ 108-116 and 118-20, could not be fair uses, then § 107 would have no function. See *generally* note 154 *supra* and accompanying text.

Professor Miller also argues that "Congress's enactment of the Semiconductor Chip Protection Act, just four years after the 1980 Software Amendments, arguably reveals something about its views regarding fair use and reverse engineering." Miller, *supra* note 162, at 1023-24. For a refutation of Professor Miller's attempt to draw a negative implication from the Chip Act and its explicit reverse engineering privilege, see Part IV(C)(2), *infra*.

301. For a discussion of copyright misuse and how the doctrine applies to contracts that violate the policy of federal patent or copyright law, or to contracts that have an anticompetitive effect in violation of federal antitrust law, see *supra* notes 185-86, 188 and accompanying text.

302. 376 U.S. 225 (1964).

303. 376 U.S. 234 (1964).

304. 17 U.S.C. § 301(a) (1988).

305. 17 U.S.C. § 301(b) (1988).

306. S. 22, 94th Cong., 2d Sess. § 301(b)(3) (1976).

307. See 122 Cong. Rec. H10910 (daily ed. Sept. 22, 1976) (revealing a concern by the Justice Department that a state law right of "misappropriation" as one which survives preemption would be a mistake).

308. U.S. Const. Art. VI.

309. See *supra* note 279.

310. See H.R. Rep. 94-1476, 94th Cong. 2d Sess. 129-131, *reprinted in* 5 U.S.C.C.A.N. 5745-5747 (1976). Since the effective date of the 1976 Act, federal copyright protection has subsisted from the moment a work was fixed in tangible form; common-law copyright protection is simultaneously preempted. *Id.*

311. *Id.* at 5747.

312. *Id.*

313. *Wright*, 953 F.2d 731 (2d Cir. 1991). See generally L. RAY PATTERSON & STANLEY W. LINDBERG, THE NATURE OF COPYRIGHT - A LAW OF USERS' RIGHTS 220 (1991):

A common practice of the producers of computer programs has been to distribute them with shrink-wrapped "licenses." These self-proclaimed licenses purport to bind the purchaser (upon opening the package) to the copyright owner's terms - which invariably restrict the use the purchaser may make of the work. Such licenses are almost surely against public policy as unilateral attempts to override public law with private law in an adhesion contract. *One can be sure that to the extent the provisions of such licenses preclude the fair use of the work, they have no legal effect*, although their *in terrorem* effect may be substantial. In our opinion, users would be well within their rights to ignore such unlawful terms and to comply instead with the law of fair use. [emphasis supplied].

314. 953 F.2d at 740.

315. *Id.*

316. EC Directive, *supra* note 3. The full text of Article 9(1) is as follows:

Article 9

Continued application of other legal provisions

1. The provisions of this Directive shall be without prejudice to any other legal provisions such as those concerning patent rights, trade-marks, unfair competition, trade secrets, protection of semi-conductor products or the law of contract. Any contractual provisions contrary to Article 6 or to the exceptions provided for in Article 5(2) and (3) shall be null and void.

Note that Article 9(1) conspicuously excepts from its rule against enforcement of contracts contrary to the rights created in Articles 5 (2) (creation of a backup copy) and (3) (reverse analysis) and Article 6 (decompilation and disassembly), any contract that is contrary to Article 5(1). Article 5 states that:

Article 5

Exceptions to the restricted acts

1. In the absence of specific contractual provisions, the acts referred to in Article 4(a) and (b) shall not require authorization by the rightholder where they are necessary for the use of the computer program by the lawful acquirer in accordance with its intended purpose, including for error correction.

2. The making of a back-up copy by a person having a right to use the computer program may not be prevented by contract insofar as it is necessary for that use.

3. The person having a right to use a copy of a computer program shall be entitled, without the authorization of the rightholder, to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he is entitled to do.

Id. Article 5(1), for example, would allow a copyright owner to contractually restrict the right of a user to load a program into a computer when the program could thereafter be used at a number of terminals.

317. 489 U.S. 141 (1989).

318. FLA. STAT. § 559.94(2) (1987), makes it unlawful "for any person to use the direct molding process to duplicate for the purpose of sale any manufactured vessel hull or component part of a vessel made by another without the written permission of that other person."

319. *Sears*, 376 U.S. 225 (1964); *Compco*, 376 U.S. 234 (1964).

320. *Sears*, 376 U.S. at 232. This federal policy is found in both the patent and copyright clause of the U.S. Constitution and the implementing legislation.

321. 379 U.S. 29 (1964).

322 *Id.* at 32.

323. *Id.* at 32-33.

324. 395 U.S. 653 (1969).

325 *Id.* at 674.

326. *Kewanee Oil Corp. v. Bicron Corp.*, 416 U.S. 470 (1974).

327 *Id.* at 493.

328. *Id.* at 485, (citing 35 U.S.C. §122).

329. 416 U.S. at 489-90.

330. 440 U.S. 257 (1979).

331. *Id.* at 264-65.

332. *Id.*

333. *Id.* at 267.

334. *Bonito Boats*, 489 U.S. at 156.

335. *Id.*

336. *Id.*

337. *Id.* at 160.

338. *Id.* (quoting *Kewanee*, 416 U.S. at 489-90).

339. See Steven W. Lundberg & John P. Sumner, *Patent Preemption of Shrink-Wrap Prohibitions on Reverse Engineering*, COMPUTER LAW., Apr. 1987, at 9; D.C. Toedt, *Bonito Boats and the Primacy of the Patent System-Are There Implications for Software Copyrights?* COMPUTER LAW., Apr. 1987, at 12; D.C. Toedt, *Bonito Boats Follow-Up: Free Competition Public Interests vs. The Substantial Similarity Test-Does the Legislative History Actually Support Nonliteral Software Copyright Protection?*, COMPUTER LAW., July 1989, at 14.; Arthur J. Levine, *Comment on Bonito Boats Follow-up: The Supreme Court's Likely Rejection of Nonliteral Software Copyright Protection*, COMPUTER LAW., July 1989, at 29. *But see* Allen R. Grogan, *Bonito Boats and Whelan: A Simple Contrast Between Patent and Copyright Protection*, COMPUTER LAW., July 1989, at 33; E. Gabriel Perle et al, *Bonito Boats Redux*, COMPUTER LAW., Feb. 1990, at 1.

340 For a discussion of the significance of these two terms, see *supra* Parts IV(D)(1)(2).

341. See 2 NIMMER, *supra* note 63, § 8.08 at 8-107, noting that under section 117 "it is only a copy made by the very act of inputting a program into a computer which is privileged." *Id.* Any subsequent retrieval of the work from the computer in a tangible form, such

as a printout, "clearly would constitute an infringing 'copy'" under both the 1909 Copyright Act and the 1976 Act. *Id.* at 8-111.

342. 17 U.S.C. § 117 (1988).

343 For the text of Article 5(3), see *supra* note 316.

344. D.C. Toedt, *Bonito Boats and the Primacy of the Patent System-Are There Implications for Software Copyrights?* COMPUTER LAW., Jan. 1989, at 12, 12.

345. *Id.* at 13.

346. *Id.*

347. *Id.* at 12.

348. *Id.*

349. *Diamond*, 450 U.S. 175, 181 (1981).

350. *Id.* at 193.

351. *Id.*

352. *See, e.g., In re Walter*, 618 F.2d 758, 767 (C.C.P.A. 1980).

353. 681 F.2d 787 (C.C.P.A. 1982).

354. *See, e.g., BENDER, supra*, note 1, § 3A.03[2], at 3A-38 (1991).

355. 684 F.2d 912 (C.C.P.A. 1982).

356 *Id.* at 913, 917.

357. 564 F. Supp. 1358 (D. Del. 1983).

358. *Id.* at 1368.

359. *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989).

360. Proprietary Rights Committee, Computer Law Section, State Bar of Michigan, *Survey of United States Software Patents Issued From July 1987 Through December 1987*, reprinted in BENDER, *supra* note 1, Chap. 3A, App. 3A[4] (1991).

361. *Id.*, BENDER, *supra* note 1, *Id.*, at App. 3A-102.

362. *Id.* at § 3A.07[3].

363. *See Survey of U.S. Business Software Patents: Post-Diehr Through December 1990*, State Bar of Michigan, 56th Annual Meeting, reprinted in 1 BENDER, *supra* note 1, App. 3A[6].

364. 35 U.S.C. § 112 (1988).

365. Compare Steven T. Naumann, *Compliance with 35 U.S.C. § 112 for Inventions Containing Computer Software: Is Disclosure of the Computer Code Required?*, 4 SOFTWARE L.J. 443 (1991) (the commentator says that the lack of disclosure is not a per se

violation of § 112, provided a person of ordinary skill in the art would have been able to practice and routinely write the program), with David Bender & Anthony R. Barkume, *Disclosure Requirements for Software-Related Patents*, COMPUTER LAW., Oct. 1992, at 1 (the commentators conclude that a detailed description of the program, including its source code, is necessary). Only a "person skilled in the art" can solve this conflict of opinions. Where such a person is enabled to rewrite the program notwithstanding the fact that the source code is not entirely described, there is no need for further disclosure. See U.S. Patent and Trademark Office, *Patentable Subject Matter-Mathematical Algorithms on Computer Programs*, Manual of Patent Examining Procedures § 2106.01 (1981).

366. Naumann, *supra* note 365.