Maximizing Protection for Computer Software

Judith A. Szepesi
I. INTRODUCTION

To shield one's home from outsiders, there are three types of protection one can use. The first and simplest type is to keep the doors and windows closed. A more elaborate form consists of locking the doors and windows. The most complex form requires the installation of a good alarm system. These forms of protection are analogous to the three types of protection one can use to safeguard software: trade secret, copyright, and patent.

Each of the forms of protection has its strengths and weaknesses and finding the optimal form is not an easy task. Trade secret protection, like keeping the doors and windows closed, is convenient but requires constant vigilance to keep others from peering inside. Copyright protection is similar to the use of a lock. The lock, however, does not protect against those who have copies of the key. Like an alarm system, a patent is an almost complete form of protection. It is, however, expensive to obtain, time consuming to install, and costly to maintain.
The attorney or legal advisor is the equivalent of a sales clerk who advises a homeowner of the optimal form of protection under the circumstances. The customer's preferences and needs must be taken into consideration when making a recommendation. Just as in the case of selecting home protection, selecting software protection is a complicated issue that is close to the hearts of many companies. There are a large number of variables that can and should be evaluated in making a recommendation.

Patent, copyright, and trade secret protection are all important, in some measure, to computer software. While trade secret was the protection of choice in the 1970s, copyright in the 1980s, and patent in the 1990s, they are all used to protect software programs today.¹ Some of these forms of protection can be used simultaneously. The advantages and disadvantages of each form of protection should be evaluated in light of the character of the company that is seeking the advice.

When assessing the need for software protection, a number of nonlegal factors should influence the practitioner's decision. The attorney must consider the period of time that the company expects to market or use the software and evaluate the financial ability of the company to pursue and enforce patents. The attorney may also assess the planned use or distribution of the software in question.

This comment will address the issues that attorneys and their clients should consider when choosing a form of software protection. It will first define key terminology with which a lawyer must be familiar to practice in this area. Second, it will discuss the three forms of protection — patent, copyright, and trade secret — and evaluate their strengths and weaknesses. Third, it will address the possibility of using these forms of protection simultaneously and the extent of protection they can provide in combination with each other. Fourth, outside factors, such as company size and economics, will be evaluated. Finally, based on all of the above information, this comment will provide a checklist to summarize these findings and make it easier to determine the optimal form of protection. It must be noted that this comment is not sufficient to make a complete evaluation of the advantages and disadvantages of the three types of available protection. A specialist should be consulted when making this decision.

---

II. Necessary Terminology

Experts have strenuously disagreed over the definitions of some of the terms in computer programming.\(^2\) It appears that terms such as "program," "algorithm," "software," "subroutine," and others are often used interchangeably and without understanding. As one commentator has noted, our legal system suffers from technical illiteracy.\(^3\) Just as law dictionaries are preferred for use in legal work, specialized dictionaries should be used for interpreting computer related language. A good understanding of computer software is necessary for any discussion of this topic.

A computer is a device capable of solving problems or manipulating data by accepting data, performing prescribed operations (mathematical or logical) on the data, and supplying the results of these operations.\(^4\) "Software" is defined by Webster’s New World Dictionary of Computer Terms as "the programs or instructions that tell a computer what to do. Software may be built into the computer’s ROM (read-only memory) or may be loaded temporarily into the computer from a disk or tape."\(^5\)

"Source code" is the symbolic code that is written by a programmer in a computer language, such as C or BASIC, and is generally decipherable by humans.\(^6\) "Object code" is the executable machine language, consisting of binary (zeros and ones) strings, that the computer actually uses to perform its actions.\(^7\) When source code is compiled, that is, translated from the computer language to the binary, it becomes object code. These definitions are generally understood by the public as well as by the courts.

"Algorithm" is the one word that causes the most confusion in understanding computers. An algorithm is defined as a "prescribed set of well-defined, unambiguous rules or processes for the solution of a problem in a finite number of steps."\(^8\) However, Webster’s New Riverside Dictionary defines it as "a mathematical rule or procedure for solving a problem."\(^9\) In Diamond v. Diehr the Supreme Court defined

\(^2\) See In Re Schrader, 22 F.3d 290, 292 n.4 (Fed. Cir. 1994).
\(^4\) WEBSTER'S NEW WORLD DICTIONARY OF COMPUTER TERMS 108 (5th ed. 1994) [hereinafter DICTIONARY OF COMPUTER TERMS].
\(^5\) Id. at 534.
\(^6\) Id. at 539.
\(^7\) Id. at 402.
\(^8\) Id. at 12.
\(^9\) WEBSTER’S II NEW RIVERSIDE UNIVERSITY DICTIONARY 92 (1988) [hereinafter WEBSTER’S].
"algorithm" as a "procedure for solving a given type of mathematical problem" and concluded that "such an algorithm, or mathematical formula, is like a law of nature which cannot be the subject of a patent." This definition has been employed by the courts since that time and should be kept in mind.

"Reverse engineering" is the technical term for taking a product or system apart in order to determine how it works. In the case of software, this usually means decompiling it. "Decompilation," also referred to as "disassembly," is translating the object code, which consists of zeros and ones, into source code, which is in human readable form. A competent programmer can take this decompiled code and use it to understand the underlying design as well as the methods used by the program.

Although the term "shrink-wrap license" is not a technical term, it is often linked to software licenses. A shrink-wrap license is a non-negotiated license with which the purchaser of almost any commercial software is familiar. It is generally packaged with the software and usually contains a number of provisions. The terms of a shrink-wrap license usually state that the software in question is not being sold, but only licensed, and they place a number of restrictions on the user. Shrink-wrap licenses generally include terms prohibiting reverse engineering and disassembly. They often contain terms limiting liability as well. Although used almost universally among software distributors, the enforceability of shrink-wrap agreements is questionable.

It must be noted that some courts use the technical terms defined above incorrectly or interchangeably. Each court decision must be ex-

12. Id. at 99, 108.
14. Id. at 51-52, 54.
15. Johnston & Grogan, supra note 1, at 5. See, e.g., Westlaw Software License Agreement (West 1995) (stating: "user may not... reverse engineer or otherwise attempt to discern the source code of the software") (on file with Santa Clara Computer & High Technology Law Journal) [hereinafter Westlaw Software License Agreement].
16. Westlaw Software License Agreement, supra note 15. Generally the publisher will only replace the defective disks, but disclaims liability for damage caused by the error. Id.
17. Agreements not to reverse engineer have not been tested in court to date. However, in two cases which tried to enforce shrink-wrap disclaimers of warranties, the clauses were held to be unenforceable. Step-Saver Data Systems Inc. v. Wyse Technology, 939 F.2d 91 (3d Cir. 1991); Arizona Retail System. v. Software Link, 831 F. Supp. 759 (D. Ariz. 1993). However, commentators have noted that because prohibitions against disassembly are ubiquitous, and because it would not adversely influence most users, they might be considered enforceable. Johnston & Grogan, supra note 1, at 10.
amined for such misuse by the practitioner before depending on its holding. The precise definition of the terms used is vital in order to understand not only the invention, but also what forms of protection may be available to such an invention.

III. PATENTS

Utility patents grant a monopoly for a limited time, allowing the patent owner to prohibit others from making, using, selling or offering for sale the patented invention in the United States. Patents can be granted only for novel, useful, and nonobvious inventions. These inventions must fall into one of the following four categories: machine, article of manufacture, process, or composition of matter. The Patent and Trademark Office (PTO) cleared up all remaining confusion about the patentability of computer software in 1989 by publishing its guidelines for software patents in its publication The Official Gazette. For a computer program to be patentable, it must, like any other invention, fall into one of these four categories.

Usefulness, in most cases, is assumed if the invention has any purpose. There are very few cases that have held that an invention was not useful. If the result produced has any application to the real world, meeting this criterion should not be a problem. In most situations, a company would not seek patent protection for an item that the PTO considers nonuseful.

Novelty is somewhat harder to evaluate, but the standard for novelty is low. The rejection of a patent application due to a lack of novelty requires a single prior art reference that discloses "each and every one of the elements" of the claimed invention. A novelty re-

18. The term of patents has recently changed as a result of the adoption of the General Agreement on Trade and Tariffs. It is now twenty years from the date of application. However, the patent is only enforceable after it issues. For patents that have an effective filing date before June 8, 1995, or that are in force June 8, 1995, the term is the longer of seventeen years from issue or twenty years from date of first filing. 35 U.S.C.A. § 154(c)(1) (West & Supp. 1995).


24. Prior art is patents, publications, and prior uses that were in existence prior to the patentee's invention or more than one year prior to filing. Mohasco Industries, Inc. v. E.T. Barwick Mills, Inc., 221 F. Supp. 191, 195 (N.D. Ga. 1963), aff'd, 340 F.2d 319 (5th Cir. 1965), reh'g denied, 342 F.2d 431 (5th Cir. 1965), cert. denied, 382 U.S. 847 (1965).

jection can generally be avoided by drafting patent claims carefully and including elements not in the prior art. If, however, the reference is for the same invention as the application, the claim usually results in an interference.\(^{26}\)

An invention must be nonobvious in order to warrant patent protection. Determining obviousness is a complex evaluation. The question of obviousness is whether “a person having ordinary skill in the art”\(^{27}\) would have been able to combine existing prior art to obtain the invention.\(^{28}\) In two decisions handed down the same day, *Graham v. John Deere Co.*\(^{29}\) and *United States v. Adams*,\(^{30}\) the Supreme Court established the rules for deciding whether an invention is nonobvious. The evaluation requires an assessment of the scope and content of the prior art, the differences between the prior art and the present invention, and the level of ordinary skill in the art.\(^{31}\) Secondary considerations such as commercial success, long felt need, and the failure of others must also be taken into account.\(^{32}\)

In *Dann v. Johnston* the Supreme Court reemphasized the provision in § 103 of the Patent Act that whether or not an invention is nonobvious is defined by what would be obvious to one reasonably skilled in the applicable art or arts.\(^{33}\) In terms of computer programs, this standard is tough to evaluate because the computer software industry is relatively young and the applicable level of skill is not well defined. This level of skill is evaluated with the use of a variety of factors, including: the educational level of the inventor, the type of problems encountered in the art, the prior art solutions, the rapidity of innovation, the sophistication of technology, and the educational level of active workers in the field.\(^{34}\) Programmers, however, are often self-taught. In fact, some of the most talented programmers are either

---

26. An interference is an action to establish which party was the first to invent, since the first inventor receives a patent. Based on 35 U.S.C. § 135, it usually involves both parties presenting documentary evidence of their invention date and diligence to the Board of Patent Appeals and Interferences. A discussion of an interference is beyond the scope of this article. For a full discussion of interferences, see 3 DONALD S. CHISUM, PATENTS: A TREATISE ON THE LAW OF PATENTABILITY, VALIDITY AND INFRINGEMENT § 10 (1995).


28. This criterion is based on 35 U.S.C. § 103.


high school or college dropouts. The standard "ordinary skill in the art" may well be defined by the skill of a self-educated young person who grew up with computers and was a hacker as a youth. The speed of innovation in this field is extremely rapid, indicating that programmers have a high level of skill. These facts complicate the determination of nonobviousness for computer programs.

Evaluating applicable prior art is complicated as well. Because the prior art in software is not well documented, and often not even available for evaluation, the patent agent's job is much tougher than in most fields. The patent attorney or agent must disclose all relevant prior art known to him and the inventor(s). This is difficult due to the lack of prior art patents, especially since the PTO only recently started to accept software patents. Most of the prior art was implemented in electronic rather than in written form. That is to say, the software engineer wrote the programs using a computer and did not document its code in paper form. As a result, inventors will face problems when choosing this method of protecting software. A competitor may have access to prior art that the inventor did not have. Because the courts or the PTO can declare patents invalid for obviousness or lack of novelty at any time, this increases the likelihood that a competitor could successfully challenge a patent based on prior art not revealed to the PTO. It could also lead to reexaminations requested by competitors, based on undisclosed prior art. An infringer can even charge the patentee with failure to disclose relevant prior art.

Even if the invention is novel, useful, and nonobvious, it must overcome one more hurdle. As the Supreme Court held in DeForest

---

35. See The Downfall of a Computer Whiz, S.F. CHRON., Feb. 17, 1995, at A1 (reporting the case of Kevin Mitnick, who was recently apprehended by law enforcement officials for committing illegal acts with the use of computers. Mitnick is a high school drop out and a reportedly brilliant programmer).
36. Wade Roush, Hackers Taking a Byte out of Computer Crime, TECH. REV., April 1995, at 32, available in LEXIS (defining "hacker" as a computer virtuoso, and as someone who wants to know the limits of computer systems and is willing to use any means to satisfy this curiosity).
38. See supra note 22 and accompanying text.
39. A patent can be declared invalid or unenforceable by a court during litigation. 35 U.S.C. § 282. The PTO can also reexamine and declare a patent invalid on someone's petition or on its own motion. 35 U.S.C. §§ 301-07.
40. Reexaminations are reviews of granted patents using patents or publications not reviewed during the prosecution of the patent. Anyone can request a reexamination upon payment of a fee. See 35 U.S.C. §§ 301-07.
41. Id.
42. Failure to disclose prior art that would be considered relevant by a reasonable examiner can invalidate a patent. 37 C.F.R.$ 1.56(a) (1994).
Radio Co. v. General Electric, laws of nature or science can never be patented. The Supreme Court held that this rule applies to mathematical algorithms, formulas, and equations, which are also unpatentable. However, the Court later explained that this only applies to its narrow definition of algorithms.

Gottschalk v. Benson was the first case that definitively evaluated the patentability of a mathematical process. In that case, the invention was a method for converting binary coded decimals (BCD) into binary numbers. The Court held that the claims merely sought to preempt the idea of this conversion. As a result, they were not patentable because they did not define any particular application for the algorithm. While ruling that the invention in question was not patentable, the Court noted that its holding did not entirely exclude computer programs from patent protection. Parker v. Flook, decided five years after Gottschalk, clarified this holding. Parker concerned the application of a newly discovered equation for updating alarm limits in catalytic conversion processes. While the Court's decision did not deal with the issue of the patentability of computer programs, it noted that mathematical algorithms, even if newly discovered, have to be evaluated as part of the prior art in evaluating novelty and nonobviousness.

The first judicial move toward allowing claims that include computer programs came in In re Bradley and Franklin. In that case, the Court of Appeals for the Federal Circuit held that firmware constitutes a machine for the purposes of interpreting 35 U.S.C. § 101, thus placing it into one of the four categories of patentable subject matter. The court noted that the important factor to evaluate is "the signifi-
cance of the data and their manipulation in the real world, i.e., what the computer is doing. However, the court specifically stated that it was not expressing an opinion on the patentability of the software itself.

In *Diamond v. Diehr* the Supreme Court finally came down on the side of software patentability. A process claim for operating a rubber molding process, using a digital computer, was upheld as patentable. The Court stated that "a claim drawn to subject matter otherwise statutory does not become nonstatutory simply because it uses a mathematical formula, computer program, or digital computer." A process claim drawn to a process executed by a digital computer could therefore be patented. The Court also noted that the patentees did not seek to preempt the use of the equation in question, but merely the use of that equation in conjunction with all of the other steps of the claimed process.

Whether a computer program consists of patentable subject matter under 35 U.S.C. § 101 is now determined by the Freeman-Walter-Abele test. As summarized in *In re Iwahashi*, this test has two steps. First, it must be determined whether the patent claim directly or indirectly recites an "algorithm" in the Benson sense of that term; a claim that fails even to recite an algorithm clearly cannot wholly preempt an algorithm. Second, the claim must be further analyzed in its entirety to ascertain whether it wholly preempts the algorithm. However, the court in *Iwahashi* then cautioned that overemphasis on the word "algorithm" was deceptive because the word only referred to the definition given to that word in Benson. The use of mathematical formulas or relationships to describe the electronic structure and operation of an apparatus does not render it nonstatutory.

A recent case that endorses patentability of software is *In re Alappat*. In that case, the Court of Appeal for the Federal Circuit held that a software program that created a smooth wave form on a digital oscilloscope was patentable. The claim was written in a

---

55. *In re Bradley and Franklin*, 600 F.2d at 812.
57. *Id.* at 187.
58. *Id.*
59. This test was developed in the cases of *In re Freeman*, 573 F.2d 1237 (C.C.P.A. 1978); *In re Walter*, 618 F.2d 758 (C.C.P.A. 1980); *In re Abele*, 684 F.2d 902 (C.C.P.A. 1982).
60. 888 F.2d at 1374.
62. 888 F.2d at 1374.
63. *Id.* at 1375.
64. 33 F.3d 1526 (Fed. Cir. 1994).
65. *Id.* at 1545.
Most importantly, the court held that "a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software."\(^6\) This allows computer software to be patented as a machine, thus falling into one of the statutory categories under 35 U.S.C. § 101.\(^6\)

Recently, the PTO has issued Proposed Examination Guidelines for Computer-Implemented Inventions.\(^6\)\(^8\) The guidelines define the following subject matter as nonstatutory: a compilation or arrangement of data independent of any physical element, a storage medium that is encoded with data representing creative or artistic expression, a data structure independent of any physical element, and a process that does nothing more than manipulate abstract ideas or concepts.\(^6\)

However, the guidelines specifically note that software could be implemented as a process, machine, or an article of manufacture.\(^6\)\(^7\) Specifically, the guidelines point out that a computer that is directed by a computer program is considered a statutory machine.\(^6\)\(^7\) Computer memory is defined as an article of manufacture.\(^6\)\(^7\) And a series of specific operational steps performed on or with the aid of a computer is a statutory process.\(^6\)\(^7\) Thus, all of these forms of claiming software inventions are considered statutory under 35 U.S.C. § 101. This approach looks at the functionality of the software and the hardware in combination, and not how the software works internally.\(^6\)\(^7\) Although these guidelines are not final, the PTO is already implementing them,\(^6\)\(^7\) which should make obtaining a software patent easier.

A patent gives the patentee the right to keep others from making, selling or using the invention, as embodied in the claims, in the United States.\(^6\)\(^7\) It does not necessarily give the patentee the right to make,
use, or sell the invention.\textsuperscript{78} The protection is quite comprehensive and protects against those who independently invent the same invention after the first inventor and those who looked at the patented item or patent disclosure to discover how it works and then make, use, or sell the invention.\textsuperscript{79}

Obtaining a patent is a costly and time-consuming process. Even if there are no problems, it can take eighteen months to three years from patent application to granted patent. If problems develop, the process can extend to over five years. If the patent is not very complex, prosecuting a patent costs approximately $5,000 to $20,000,\textsuperscript{80} but if problems develop, this figure can increase dramatically. In addition, a qualified patent agent must be found to prosecute the patent.\textsuperscript{81}

The working software community, especially programmers, often have an anti-patent bias.\textsuperscript{82} Only about 4,000 software patents were issued by 1994.\textsuperscript{83} However, the use of software patents is increasing, especially by large companies. Today, software patents cover such things as system functions, mouse control, speech recognition, and display functions like computerized clipboards.\textsuperscript{84} Since the Court of Appeals for the Federal Circuit was created, significantly more patents have been held to be valid.\textsuperscript{85} Enforcement of software patents, although extremely costly, is usually successful, which in turn increases the value of software patents.

Once a patent is granted it is presumed to be valid.\textsuperscript{86} This presumption is based on the extensive examination conducted during patent prosecution. Although the duration of the patent is only twenty

\textsuperscript{78} This could happen if the product would infringe on other patents. In such cases, the patent owners generally cross-license each other, so that they may use each other's patents.
\textsuperscript{79} 35 U.S.C. § 271.
\textsuperscript{81} Although an inventor can chose to proceed alone, this is generally not advisable, especially in a complicated area, such as software.
\textsuperscript{82} Many seem to feel that their work should be available for others to use and to improve on. Paul Heckel, \textit{The Software-Patent Controversy}, \textit{Computer Law.}, Dec. 1992, at 13.
\textsuperscript{84} See e.g., U.S. Pat. Nos. 5,443,068; 5,440,663; 5,442,742; 5,394,546.
\textsuperscript{85} The Court of Appeals for the Federal Circuit was created in 1982. The Federal Circuit judges generally find patents valid more often than the other circuits. Since they specialize in patent matters they also appear to have a better understanding of the underlying technical issues. Today more than half of challenged patents are held valid. Ludwig E. Kolman & Thomas L. Duston, \textit{In Patent Litigation, Can Teamwork Work?}, Nat’l L. J., Jan. 24, 1994, at S15.
\textsuperscript{86} 35 U.S.C. § 282.
years from the original filing, the protection during that time is almost all-inclusive. Although difficult to accomplish, it is possible to design around a patent. By leaving out elements that are claimed in the patent, one can generally avoid infringement. However, this is close to impossible if the patent claims are well drafted and include only essential terms.

Just as a house containing many valuables needs to be protected by an expensive alarm system, patents offer the most effective form of protection for valuable software.

IV. COPYRIGHTS

Copyright protection is granted to "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." Ideas, processes, or methods of operations are not protected by copyright.

In Apple Computer, Inc. v. Franklin Computer Corp., the Third Circuit Court of Appeals held that computer programs are copyrightable. The court upheld a copyright granted on the source and object code of a computer program. It noted that it was irrelevant for evaluating copyright validity whether the program was fixed in memory or on a ROM, or whether it was an application program or an operating system. The court also held that copyright protection extends only to the expression embodied in the program, not to the underlying processes.

Section 101 of the Copyright Act now includes the definition of a computer program. In Apple Computer, Inc. v. Formula Int'l, Inc.

87. The term was recently changed by the adoption of the General Agreement on Trade and Tariffs. For any patent filed after June 8, 1995, the term is 20 years from date of the earliest application filing. 35 U.S.C.A. § 154 (West Supp. 1995). See generally, Patricia Montalvo, How Will the New Twenty-Year Patent Term Affect You? A Look at the TRIPs Agreement and the Adoption of a Twenty-Year Patent Term in this issue.
92. 714 F.2d at 1249.
93. Id. at 1249, 1252.
94. Id. at 1253.
95. "A 'computer program' is a set of statements or instruction to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101.
96. 725 F.2d 521 (9th Cir. 1984).
the Ninth Circuit Court of Appeals held that § 101 provides copyright protection for the expressions contained in a computer program.\footnote{77} Copyright protects only expression, and not the idea, as codified in 17 U.S.C. § 102 (b). However, the line between copyrightable expression and an idea that cannot be copyrighted is impossible to draw with precision.\footnote{78} This problem is further complicated because computer programs are utilitarian works.\footnote{79} Software is designed to accomplish a task and most people do not think of it as a literary and creative work. In \textit{Baker v. Selden},\footnote{100} the Supreme Court drew these lines when it stated that "copyright of a work on mathematical science can not give to the author an exclusive right to the methods of operation he propounds, or the diagrams which he employs to explain them."\footnote{101} Because the utility of a computer program is intertwined with its expression, the courts have struggled with separating these two concepts.

The merger doctrine also complicates the issue of software copyright protection. This doctrine stands for the proposition that if there are only a limited number of ways to express an idea, then copyright cannot be granted to any of them.\footnote{102} This prevents the first few authors from monopolizing the whole field. No court has ever defined this "limited number." This causes problems when copyrighting software because some functions executed by software only have a limited number of ways in which they can be implemented. Courts have used the merger doctrine to eliminate certain portions of software from copyright protection. In \textit{Johnson Controls, Inc. v. Phoenix Control Systems, Inc.}, the Court of Appeals for the Ninth Circuit explained that when an idea and its expression become inseparable, the expression cannot be given copyright protection, even if it were otherwise eligible for protection.\footnote{103} This standard, first set out by Judge Browning in \textit{Herbert Rosenthal Jewelry Corp.},\footnote{104} which was a case regard-

\footnote{77} Id. at 525.
\footnote{78} Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), \textit{cert. denied}, 282 U.S. 902 (1931).
\footnote{79} Courts have held in the past that purely utilitarian works are not copyrightable and that if the work had utilitarian as well as artistic aspects, only the artistic elements were protected. Mazer v. Stein, 374 U.S. 201, 218 (1954). However, no court has ever held that software cannot receive copyright protection on this basis.
\footnote{100} 101 U.S. 99 (1879).
\footnote{101} Id. at 103.
\footnote{102} It is based on 17 U.S.C. § 102(b). \textit{See Gates Rubber Co. v. Bando Chemical Indus.}, 9 F.3d 823, 838 (10th Cir. 1993).
\footnote{103} 886 F.2d 1173, 1175.
\footnote{104} Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971) (holding that there were only a limited number of ways a jewel encrusted bee could look, and therefore the idea of a jewel encrusted bee merged with its expression, and precluded protection).
ing the design of jewel encrusted figurines of bees, has been the touchstone of the limits of software protection by copyright.

The nonliteral elements of software are also protected to some extent.\textsuperscript{105} Nonliteral elements, like a detailed plot outline of a play, have been granted copyright protection in the past.\textsuperscript{106} Nonliteral elements of computer software, such as the detailed flowchart and structure, have been granted some measure of protection as well, though the courts are split about the extent of this protection.\textsuperscript{107} The minority position was expressed in \textit{Whelan Assoc. v. Jaslow Dental Lab., Inc.},\textsuperscript{108} where the Court of Appeals for the Third Circuit extended protection to the structure, sequence, and organization of the computer program.\textsuperscript{109} Most other circuits followed the abstraction-filtration-comparison test first set out by Judge Learned Hand.\textsuperscript{110} This method was clarified with respect to computer programs\textsuperscript{111} in \textit{Computer Associates v. Altai}.\textsuperscript{112} The abstraction step entails dividing the program into varying levels of abstraction starting with the general idea and going down to its specific implementation. The filtration examines each of these levels and discards those that cannot be protected. That is, the main idea and the general concepts behind the program cannot be protected by copyright. The rest of the program is then dissected into component parts. Those that are not considered copyrightable are filtered out.\textsuperscript{113} Elements dictated by efficiency are filtered out because there are only a limited number of ways to create an efficient program. Elements dictated by external factors, such as the other software or hardware with which the program interfaces, are also filtered out because these interfacing techniques are considered standard. In fact, they are often compared to stock characters in plays.\textsuperscript{114} Finally, elements

\textsuperscript{105} Examples of software nonliteral elements are the organization, general function and other aspects of the program that are not part of the actual code written by the programmer.

\textsuperscript{106} Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), cert. denied, 282 U.S. 902 (1931).


\textsuperscript{109} Id. at 1248.

\textsuperscript{110} Nichols, 45 F.2d at 121 (holding that for a comparison of similarities between plays an abstraction-filtration test has to be used).


\textsuperscript{112} 982 F.2d 693 (2nd Cir. 1992).

\textsuperscript{113} Id. at 714.

\textsuperscript{114} Id. at 709.
taken from the public domain are filtered out. Just as with patent, copyright protection cannot remove something from the public domain; factors taken from public domain are not the original expression of the author, and thus cannot be protected. The remaining portions of the software program are considered protectable under the copyright laws. The question as to which nonliteral elements are protected is highly fact specific and evaluated separately in each situation.

Thus, the amount of protection provided by copyright largely depends on the type of software being protected. Generally, interface software, such as printer or modem drivers, would not receive much protection from copyright law. Even operating systems may not be protected to a great extent since many of their functions are dictated by the hardware. Application programs are more protected, especially programs that incorporate creative and expressive elements beyond their functionality. A new game would probably receive more value from copyright protection than a word processing program that uses the same functions as all other word processing programs on the market.

Copyright gives the copyright holder the exclusive right to make copies, prepare derivative works, and distribute copies of the work. However, it is not infringement to examine a copyrighted work and to create a similar work that achieves the same goals, so long as expressive elements are not copied.

Copyright law generally prohibits disassembly. Disassembly itself creates a number of copies, and the printout of the results of the decompilation can be considered an unauthorized copy. However, Sega Enterprises, Ltd. v. Accolade, Inc. notes that disassembly is acceptable under the "fair use standard" to discover unprotected and unprotectable parts of the software. Fair use is an equitable defense evaluated on a case-by-case basis. Those trying to reverse engineer must justify their actions under this standard. The problem of disassembly can be avoided by licensing software instead of selling it since software licenses include a provision prohibiting disassembly.

The reader should note that this concept is changing. The Copyright Act includes a new section that will restore copyright in certain works that were formerly in the public domain in the United States. 17 U.S.C.A. § 104A (West Supp. 1995). Computer Assoc. Int'l, 982 F.2d at 710.

Copyright does not protect the underlying ideas and processes. 17 U.S.C. § 102(b).


Copyright does not protect the underlying ideas and processes. 17 U.S.C. § 102(b).

See also supra note 11 and accompanying text.

Sega Enterprises, Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520 (9th Cir. 1992).

Johnston & Grogan, supra note 1, at 8.
It is fairly simple to receive protection under the Copyright Act. Copyright protection for works created on or after January 1, 1978, occurs at the time of fixation in a tangible medium from which it can be reproduced. The term of copyright is either (a) the life of the author plus 50 years or (b) if it is a work for hire or the author is a corporate entity, 75 years from first publication or 100 years from creation, whichever expires first. Neither publication nor registration is necessary in order to receive this protection. No investment beyond the creation and fixation of the work is required for copyright protection. However, registration is a prerequisite for filing suit in federal court. Special damages can also be collected if the copyright is registered.

Copyright automatically attaches to any work fixed in a reproducible medium. This definition includes software, but the extent of the protection depends on the factors enumerated above. Therefore, an attorney should evaluate the type of software sought to be protected and determine whether additional measures are necessary to provide sufficient protection. Just as using locks on one's door is almost an automatic first level of defense, copyright is a very useful first step. Yet it may not always be enough because many aspects of software may not be protected.

V. TRADE SECRETS

Software has long been considered one of the most valuable assets that can be protected by trade secret law. The protection of software by trade secret law has received very little attention in the last decade. However, the recent verdict in Stac Electronics v. Microsoft Corp. has again placed this issue on the agenda of companies involved in the computer industry.

103. Id.
104. Id. § 408.
105. Id. § 102(a).
106. Id. § 412. This rule applies unless the work was created in a country outside of the United States that is a signatory to the Berne Convention. Id.
107. Id. § 412. If a copyright is registered before the start of infringement or less than three months from creation, statutory damages and attorney's fees are recoverable. Id.
108. Id. § 102(a).
110. No. 93-0413 (S.D. Cal. decided Feb. 23, 1994). Pursuant to the settlement agreement between the parties, all orders, verdicts, and judgments in the case have been vacated. Stac Elecs. v. Microsoft Corp., 38 F.3d 1222 (Fed. Cir. 1994).
111. Johnston & Grogan, supra note 1, at 1.
Trade secrets can protect every aspect of software if it has a minimum amount of novelty.\textsuperscript{132} To merit protection, the item must merely differ to a small degree from items commonly available to the public.\textsuperscript{133} Both underlying ideas and processes, as well as specific expressions used, are considered within the penumbra of trade secrets.\textsuperscript{134} Unlike patents, novelty and uniqueness are not requirements for trade secret protection.\textsuperscript{135} Trade secrets, unlike patents and copyrights, are protected by state statutes.\textsuperscript{136} Yet the protection afforded trade secrets is virtually identical in all states.\textsuperscript{137} Most states base their trade secret legislation on either the Uniform Trade Secrets Act (UTSA) as amended in 1985\textsuperscript{138} or the Restatement of Torts.

The Uniform Trade Secrets Act defines a trade secret as "information, including any formula, pattern, compilation program, device or process, which derives independent value from not being generally known and is the subject of reasonable efforts to maintain its secrecy."\textsuperscript{139} The Restatement of Torts offers an alternative definition: "A trade secret may consist of any formula, pattern, device or compilation of information which is used in one's business and which gives him an opportunity to obtain an advantage over competitors who do not know or use it."\textsuperscript{140} As the comment to the Restatement observes: "[A] substantial element of secrecy must exist, so that, except by the use of improper means, there would be difficulty in acquiring the information."\textsuperscript{141} The Restatement also enumerates some factors that determine whether a particular piece of information is considered a trade

\textsuperscript{133} Id.
\textsuperscript{134} Computer Assoc., Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 717 (3d Cir. 1993).
\textsuperscript{137} Albert C. Smith & Jared A. Slosberg, Beware! Trade Secret Software May be Patented by a Later Inventor, COMPUTER LAW., Nov. 1990, at 15.
\textsuperscript{138} Thirty-eight states have adopted the Act in some form. Every state that adopts a form of the UTSA must have its statute approved by the National Conference of Commissioners on Uniform State Laws. For a complete list with citations, see Paul Goldstein, et al., SELECTED STATUTES AND INTERNATIONAL AGREEMENTS ON UNFAIR COMPETITION, TRADEMARK, COPYRIGHT AND PATENT 16-17 (1994).
\textsuperscript{139} UNIFORM TRADE SECRETS ACT § 1(4) (1985).
\textsuperscript{140} RESTATEMENT OF TORTS § 757, cmt. b (1939). The Restatement of Unfair Competition has superseded this section of the Restatement of Torts. However, its definitions are still used by courts today.
\textsuperscript{141} Id.
secret or not. Actual trade secret status, however, is determined by a court during litigation, which can be used to enforce trade secrecy or to recover damages for a violation of trade secret laws.

Although trade secret protection can attach to almost anything, it only protects against misappropriation. In *Bonito Boats v. Thunder Craft* the Supreme Court noted that reverse engineering on materials available to the public is not prohibited even though it may contain elements protected by trade secret. That is, once a product is placed on the market, it becomes available to one's competitors and, unless other measures are taken, these competitors are not prohibited by trade secret law from reverse engineering, copying, or using the software. Decompilation is fair game unless it is prevented in other ways. It is possible to optimize the code in a way that will foil most currently existing decompilers, although a good software engineer could nonetheless decompile the program and end up with code similar to the original source code.

Courts have held that if reverse engineering would be time consuming and expensive, trade secrecy can exist even when a product is

---

142. *Id.* These factors are: "(1) the extent to which the information is known outside of his business; (2) the extent to which it is known by employees and others involved in his business; (3) the extent of measures taken by him to guard the secrecy of the information; (4) the value of the information to him and to his competitors; (5) the amount of effort or money expended by him in developing the information; (6) the ease or difficulty with which the information could be properly acquired or duplicated by others." *Id.*


144. (2) "Misappropriation" means:

(i) acquisition of a trade secret of another by a person who knows or has reason to know that the trade secret was acquired by improper means; or

(ii) disclosure or use of a trade secret of another without express or implied consent by a person who

(A) used improper means to acquire knowledge of the trade secret; or

(B) at the time of disclosure or use, knew or had reason to know that his knowledge of the trade secret was

(I) derived from or through a person who had utilized improper means to acquire it;

(II) acquired under circumstances giving rise to a duty to maintain its secrecy or limit its use; or

(III) derived from or through a person who owed a duty to the person seeking relief to maintain its secrecy or limits its use; or

(C) before a material change of his position, knew or had reason to know that it was a trade secret and that knowledge of it had been acquired by accident or mistake.

(3) "Person" means a natural person, corporation, business trust, estate, trust, partnership, association, joint venture, government, governmental subdivision or agency, or any other legal or commercial entity.

*Uniform Trade Secrets Act* § 2.


146. *Id.* at 155. *See also Kewanee Oil Co.*, 416 U.S. at 476.
placed on the market.\textsuperscript{147} In fact, merely because one person reverse engineers the product does not necessarily defeat trade secrecy.\textsuperscript{148} It is notable that many experienced programmers acknowledge that systematic disassembly and reverse engineering of program code for purposes of understanding the internal design of a competitor’s program is a relatively rare and unseemly practice.\textsuperscript{149} In fact, the defense of reverse engineering is often theoretical; a defendant will assert that no trade secrecy can exist because the code could easily have been reverse engineered even though it was not in fact done.

In \textit{Televation Telecom Sys. v. Saidon}\textsuperscript{150} the court held that although a competitor could duplicate a plaintiff’s product through reverse engineering, the existence of a trade secret was not precluded, particularly where the evidence demonstrated that the reverse engineering process would be time consuming.\textsuperscript{151} Of course, this only applies when the defendant did not conduct reverse engineering but obtained the trade secrets in some other manner. It means that the mere possibility of reverse engineering does not preclude trade secret protection.

If a product or process is substantially derived from a trade secret, the person who misappropriated it cannot use it even with independent improvements.\textsuperscript{152} Software developers may market their programs with a license that includes a provision prohibiting decompilation in an effort to prevent reverse engineering.\textsuperscript{153} When such a license is individually negotiated and both parties sign it, it would most likely be enforceable. Thus, anyone who used such licensed software outside the bounds of the license would be violating trade secret law. However, the validity of shrink-wrap licenses has not been fully litigated with respect to this issue. Therefore, it may be unwise to rely on trade secret law for mass-marketed software.

Trade secrets are neither registered nor recorded. Yet the trade secret holder must demonstrate a continuing course of conduct to protect the secret.\textsuperscript{154} In litigation, the plaintiff must be able to produce documentation that a trade secret existed and that the company pro-

\begin{thebibliography}{154}
\bibitem{147} See Johnston & Grogan, supra note 1, at 7.
\bibitem{148} CAL. CIV. CODE § 3426.1 LEGIS. COMM. CMT. (1984).
\bibitem{149} Johnston & Grogan, supra note 1, at 6.
\bibitem{151} Id. at 1365.
\bibitem{153} Smith & Slosberg, supra note 137.
\bibitem{154} Electro-Craft Corp. v. Controlled Motion, Inc., 332 N.W.2d 890, 901 (Minn. 1983).
\end{thebibliography}
tected the trade secret. To prove that adequate measures were taken to protect secrecy, nondisclosure agreements with employees may be required. Some cases have held that all employees must sign nondisclosure agreements, while others have concluded that it is sufficient if all persons in contact with the trade secret are aware of its confidentiality. In Jostens, Inc. v. National Computer Systems, Inc., the Minnesota Supreme Court found that by allowing prospective customers to tour the area in which the processes protected by trade secret were used, plaintiffs showed a lack of concern with confidentiality, which contributed to defeating the claim of trade secrecy. The exact extent of "reasonable efforts" is evaluated on a case-by-case basis, but this does put a substantial burden on the individual trying to protect a software program or system by trade secret.

Additionally, there is a danger that a later inventor may patent software that has been held as a trade secret by the original inventor. If the second inventor gets a patent on the invention, the first inventor will be barred from making, using, or selling the patented invention. The first inventor is barred from applying for a patent if the patented invention or process has been used for more than one year. This "on sale bar," however, does not apply to the second inventor who developed the invention while the first invention was held as a trade secret. Therefore, in contesting a patent granted to a second inventor, the first inventor is in the uncomfortable position of trying to prove that the invention he or she tried to keep as a trade secret was in fact publicly known.


156. See Pressure Science Inc. v. Kramer, 413 F. Supp. 618, 626-27 (D. Conn. 1976) (holding that plaintiff's failure to require all employees working in the confidential area to sign nondisclosure agreements was evidence of a lack of concern for confidentiality, and defeated his claim of existence as a trade secret), aff'd without op., 551 F.2d 301 (2d Cir. 1976).


159. 318 N.W.2d 691 (Minn. 1982).

160. Id. at 700. Other factors included a technical article presented by a Jostent's employee and the fact that nothing was marked secret. Id.


162. 35 U.S.C. § 154 sets out the scope of patent protection.

163. 35 U.S.C. § 102(b) (barring patentability if the invention had been in public use or sale for more than one year prior to the date of the patent application).
In one well-known case, *W. L. Gore & Assocs. v. Garlock, Inc.*, the first inventor of a machine produced what is now known as Gore-tex by stretching unsintered polytera-flouorethylene (Teflon) to ten times its original size; he sold the machine under a confidentiality agreement. The second inventor, Dr. Gore, filed a patent application on the process and product two years later. After the patent was granted, Dr. Gore sued Garlock for patent infringement and Garlock argued that there was prior use and sale. The court held that the Gore patent was valid, the “activity having been secret, not public.” It is unlikely that the first inventor who keeps his or her invention under trade secret protection can invalidate a patent granted to a later inventor. In fact, the first inventor will most likely be barred from marketing or licensing the invention. This danger discourages maintaining software as a trade secret in areas in which the companies have high research and development costs and are close to each other in their respective levels of technological innovation.

Trade secret protection is considered fragile because there is no recapturing a secret once it has been released. For example, if an employee posted the source code of a trade secret software program on the Internet, the trade secret is lost. Once a trade secret becomes public, it is unlikely to be recovered. Even if the person who revealed the trade secret was in violation of employment contract obligations, only damages remain to compensate the former trade secret holder. If the person who discovers the secret does not publish it, but merely tries to sell it or use it, an injunction may recover the secret and prevent its misappropriation. The danger, of course, is that one person can deprive a company of its trade secrets. In seeking software protection, the practitioner should contemplate whether the possible loss of the trade secret is a risk that one can take.

A trade secret is created when the inventor takes an acceptable level of precautions to protect the items considered secret. A trade

---

165. 721 F.2d at 1549.
166. *Id.* at 1546.
167. *Id.* at 1549.
168. *Id.* at 1550 (Hence, the fabric is named Gore-tex after the inventor whose patent granted him a 17 year monopoly in making the material.).
171. See, e.g., UTSA, *supra* note 139, at § 2; CAL. CIV. CODE § 3426.2 (1985).
172. See *Restatement of Torts* § 757, com. b (1939); *Uniform Trade Secrets Act* § 1(4).
secret has no set duration, and it generally lasts until the secret becomes known. Usually if a product or process that incorporates, but does not disclose, a trade secret has been sold or publicly used for a long period of time, the trade secret can be considered expired. However, some trade secrets can be available yet protected for hundreds of years. Thus, just as closing one's doors and windows can be an effective defense in a limited number of cases, trade secret can protect every aspect of software in a limited number of situations. However, one peek in the window can destroy the fragile trade secret protection.

VI. Coexistence and Mutual Exclusivity

The sections above describe the various forms of protection that may be used for computer software. Some of these forms are mutually exclusive, while others can peacefully coexist or even enhance each other. Patent and copyright protection can coexist in some instances; different aspects of the same computer program or software package can be protected by either copyright or patent. In fact, patent law now allows a copyright symbol, ©, on the drawings illustrating a patent application. Although the Copyright Office has held that copyright registration will be denied to objects that have been granted a design patent, the General Counsel to the United States Copyright Office has stated that this regulation does not apply to computer programs protected by utility patents.

A number of commentators have taken sides on this issue, and the issue has not been settled. To date there are no cases in which an inventor applied for both utility patent protection and copyright protection and someone challenged the coexistence of the two forms of protection. Statutorily, both copyright and patent preempt state laws that provide identical protection, but neither of them preempts

174. Coca-Cola is a prime example. The actual formula for Coke has been a closely held trade secret for over 100 years, even though Coca-Cola is commercially sold.
175. MANUAL OF PAT. EXAMINING PROCE. § 608.01(v) (1995).
176. 37 C.F.R. § 202.10(b) (1994) (Note that this only denies copyright registration once the patent has been granted, if the application for copyright is before application or grant of patent, copyright registration will not be refused.).
178. See Michael J. Kline, Requiring an Election of Protection for Patentable/Copyrightable Computer Programs, 6 COMPUTER/L.J. 607, 638-75 (1986) (suggesting that one or the other form of protection must be elected). But see Einhorn, supra note 177, at 274 (concluding that there is no reason why copyright and patent protection cannot coexist).
other federal statutes. There is no solution to this problem in either the patent or copyright acts. Because this is still a growing area of the law, this issue may be litigated fairly soon. For now, applying for patent protection while registering a work for copyright is a reasonable option.

Patent and trade secret protection are generally mutually exclusive. The purpose of patent protection is to reveal innovations, so that after the patent term has expired, the invention can be placed in the public domain. A trade secret, once revealed, loses all of its effectiveness, while in order for a patent to be valid it must disclose the best mode of implementing the invention.

While the two types of protection appear clearly incompatible, there is one instance in which a trade secret element can be part of a patented program. If equivalent elements are known and available to those skilled in the art, one may refer to a trade secret element by name only. For example, one may refer to an ingredient in a mixed drink such as Coca-Cola without revealing the actual chemical composition of Coca-Cola, because there are known equivalents available on the market and a reasonable amount of experimentation can produce a suitable substitute. In the case of computer programs, a trade secret subroutine can be referred to by name, if those of "ordinary skill in the art" are aware of an equivalent, such as an alternative implementation. However, if the trade secret is the heart of the patented program it must be disclosed in order to have a valid patent. For example, the patent in *White Consolidated Industries v. Vega Servo Control, Inc.* was directed to a computer program that controlled machine tools. The program allowed the operator to interact dynamically with the machine using an English-like language that was translated into machine readable form by a translator referred to as "SPLIT." The court held the patent invalid for failure to disclose the best mode of operation by maintaining SPLIT as a trade secret.

---

181. A trade secret, on the other hand, has to be concealed from the public to be valid. *Id.*
185. 713 F.2d 788 (Fed. Cir. 1983).
186. *Id.* at 789.
187. *Id.* at 791.
In the alternative, in a large program a small innovative subroutine or segment can be protected by patent while the rest of the program, and the way it interrelates, remains protected by trade secret. Finally, while a patent application is pending, it is held in secrecy; if a patent never issues, its contents can remain a trade secret.188

Copyrights and trade secrets are not mutually exclusive, although copyright law preempts all state laws that offer identical protection to the federal copyright act.189 The “extra element test” requires that a trade secret be differentiated from copyright in some way in order to avoid preemption.190

The Code of Federal Regulations concerning deposit for copyright registration has a section dealing with trade secrets. The regulations state that material being registered may contain redacted trade secrets if the remaining portion contains “an appreciable amount of original computer code.”191

A series of cases have held that the requirement of breach of trust or confidentiality, which is a prerequisite for an action in trade secret misappropriation, is a valid extra element.192 In Comprehensive Technologies International, Inc. v. Software Artisans, Inc.,193 the Court of Appeals for the Fourth Circuit held that computer programs can qualify for both forms of protection. Computer programs can simultaneously constitute literary works under federal copyright law and processes or programs under state trade secret law.194 Because there is no preemption, trade secret and copyright protection can coexist. In fact, trade secret and copyright law together may provide a high degree of protection. Copyright law generally prohibits disassembly because it involves the making of a number of copies,195 which may prevent competitors from getting access to those parts of the program that are trade secrets. The prohibition of copying imposed by copyright makes disassembly of trade secret programs much more compli-

188. 35 U.S.C. § 122. This may change in the future if the United States decides to follow other countries and publish applications 18 months after filing, thus destroying any trade secrets included in the patent disclosure.
189. 17 U.S.C. § 301 preempts all state laws that are equivalent to the exclusive rights protected by copyright. However, if the legal or equitable rights are not equivalent to any of the exclusive rights within the scope of copyright, there is no preemption.
193. 3 F.3d 730 (4th Cir. 1993).
194. Id. at 736 n.7.
195. Sega v. Accolade, 977 F.2d 1510, 1519 (9th Cir. 1992).
CATALOG 1996

PROTECTING COMPUTER SOFTWARE

Sega Enterprises, Ltd. v. Accolade, Inc.\textsuperscript{196} discusses the right to disassemble. The court in that case held that disassembly of object code may fall within the "fair use" defense, if it is only to obtain information that is not protected by copyright.\textsuperscript{197} The intermediate copies were allowed under the fair use doctrine in copyright law to discover material that was unprotected by copyright. However, it is possible that the discovered material, though uncopyrightable, is claimed as a trade secret. If the reverse engineering process is difficult and time consuming, or if the material is covered by a negotiated license prohibiting reverse engineering, the trade secret may be protectable despite its uncopyrightable nature. One commentator, however, suggested that if the entire item is protected by trade secret, there may not be any unprotected information.\textsuperscript{198} This theory, though advocated by some commentators,\textsuperscript{199} has to date not been tested in the courts.

VII. NONLEGAL FACTORS TO BE CONSIDERED IN OBTAINING SOFTWARE PROTECTION

The above discussion introduces the legal concerns that must be taken into account when choosing a form of software protection. However, there are nonlegal issues that affect the level and usefulness for the various forms of protection which must be looked at as well.

How the software was developed is an issue that has legal implications vital in selecting the form of protection available and valuable to the software in question. For example, software that is not novel and nonobvious does not receive patent protection.\textsuperscript{200} Information that may have been complicated to collect, such as some database entries, may not be copyrightable or patentable.\textsuperscript{201} Therefore, the development process of the software has to be examined when determining which type or types of protection to select.

\textsuperscript{196} 977 F.2d 1510 (9th Cir. 1992).
\textsuperscript{197} Id. at 1520.
\textsuperscript{198} Johnston & Grogan, supra note 1, at 8:

\textit{Sega} did not involve a program in which trade secrets were claimed. The court of appeals emphasized in \textit{Sega} that disassembly was the only way to achieve access to "unprotected" aspects of Sega's program. Therefore, \textit{Sega} may not permit access to internal aspects of a program protected as a trade secret.

\textit{Id.} (citing Sega v. Accolade, 977 F.2d 1510, 1520 (9th Cir. 1992)).
\textsuperscript{199} Id.
\textsuperscript{200} See supra notes 18-42 and accompanying text.

The anticipated use of the software is another important nonlegal consideration. If the software is to be used internally, it should be treated differently than software that will be sold to the general public. Trade secret laws are more likely to protect software used inside a company. Because only a controlled group has access to the software, the problem of reverse engineering does not need to be addressed. Patent protection is not very useful in such an instance, since it would be almost impossible to determine whether a competitor had an infringing product. The large investment required to obtain a patent also weighs against patent as a choice.

Commercial considerations must also be taken into account. If the company is a small startup, it may need a large patent or copyright portfolio for investment purposes. It may be worthwhile to pursue patents and copyrights even if the other considerations appear to weigh against it.

The commercial viability of the product also has to be taken into account. The average shelf-life of a new software product is estimated at nine months. Since the average time to obtain a patent is over eighteen months and patents provide limited protection before they are granted, such software may not be optimally protected by patent. Many software products depend on their predecessors, however. Thus patenting the original product may lead to protection for further developments. If the product or operating system becomes a de facto standard, it, or at least the underlying concepts, may last well beyond the nine-month average shelf life.

The ability of the company to provide some sort of trade secret maintenance program must also be considered. Commentators recommend that a company that wishes to maintain trade secrets do a yearly audit of the information that is considered secret. Such an inventory requires a review of all the products and information used by the company, its value, and trade secrecy status. Having an inventory can be very helpful in a litigation situation, as well as in making em-

203. For example, in the Apple v. Microsoft litigation the license granted by Apple to Microsoft for the first version of Windows was used to protect further versions of Windows from being considered infringing devices. The same principles apply to patent protection. See Apple Computer Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), cert. denied, 115 S. Ct. 1176 (1995).
204. See for example DOS (Disk Operating System) which was created by Microsoft in the early 1980s. The succeeding versions of DOS incorporated many of the functions found in the original version.
206. Id. at 975.
ployees aware of the trade secret status of the information. Patents also require some maintenance; however, this maintenance is merely monetary. After obtaining a patent, the patent owner must pay increasing maintenance fees 3, 7 and 11 years after the patent was issued. The ability to continue such a consistent maintenance should be evaluated. If the company is unlikely to monitor its trade secrets carefully, they may not exist as secrets for very long.

The cost of obtaining protection can also be a significant issue, especially for smaller firms. The cost to obtain protection for software varies widely. A utility patent application on a fairly simple invention has a median cost of $4,010. For the original application on a fairly complex computer software invention, the median cost is $6,980. For basic prosecution of the patent, which includes filing an information disclosure statement, amendments and general arguments the median cost is $1,233; for more complex software prosecutions the median cost is $1,602. If an appeal is required, median costs will increase by $2,775.

While a copyright registration costs $20, to have an attorney prepare a copyright application costs approximately $200. However, a copyright does not need to be registered to be valid. In fact, copyright registration is often only filed when infringement is suspected. Since trade secrets are neither registered nor recorded, such costs do not apply. However, maintaining secrecy can be very expensive.

The ability of the company to support such expenditures has to be evaluated. If the company is small, the number of patents sought will be small and the worth of the patent will have to be carefully evaluated. If the company is large and financially sound, such considerations may not be that important.

Litigation costs vary widely. Patent litigation is the most expensive. A patent infringement suit has a total median cost through discovery of $500,000. The total median cost through trial is

---

207. Id.
209. AM. INTELL. PROP. L. ASSOC. COMMITTEE ON ECON. OF LEGAL PRAC., REP. OF ECON. SUR. 1995, at 56 [hereinafter AIPLA REPORT].
210. Id. at 56.
211. Id. at 57-8.
212. Id. at 58.
213. Id. at 60.
215. Copyright registration is necessary for litigation (17 U.S.C. § 411), and therefore, once copyright infringement is suspected, registration is generally effected quite quickly.
216. AIPLA REPORT, supra note 209, at 61.
Copyright infringement suits are less expensive. The median cost of a copyright infringement suit is $100,000 through discovery and $200,000 through trial.\textsuperscript{218} As some commentators have noted, the cost of copyright infringement suits is rising with the increased necessity to use expert witnesses and the increased complexity of the issues being litigated.\textsuperscript{219} The median cost of a trade secret misappropriation suit is $201,000 through discovery and $355,000 through trial.\textsuperscript{220}

If more than one form of intellectual property is involved in the trial, such as a combination of copyright and trade secret, or patent and copyright, the median cost through discovery is $498,000.\textsuperscript{221} With a trial, this figure increases to $752,000.\textsuperscript{222} It must be noted that a suit involving copyright and trade secret would probably cost much less than one involving patents and trade secrets.

There are alternatives to litigation for copyright, patent and trade secrets. Arbitration, mediation, and mini-trials have been employed by attorneys.\textsuperscript{223} The cost of these alternative dispute resolution methods is much lower than in conventional litigation. Estimated median costs for alternative dispute resolution are: $50,000 for mediation; $78,000 for med/arb; $100,000 for mini-trial; $150,000 for summary jury trial; $151,000 for binding arbitration.\textsuperscript{224} It must be noted that over 90 percent of cases settle well before the end of trial.\textsuperscript{225}

The willingness and ability of the firm to pay a large investment in enforcement should be examined. If it is unlikely that the company will be willing to invest enough money to enforce a patent, a patent may not be an optimal choice. Note that combining forms of protection will probably make the litigation more expensive than if only one form of protection is involved.

There are only minor variations between the remedies available in patent, copyright, and trade secret litigation. Patent remedies include injunctions against further use of the patented article, as well as damages.\textsuperscript{226} The damages awarded should be adequate to compensate

\textsuperscript{217} Id.
\textsuperscript{218} Id. at 62.
\textsuperscript{219} A. Samuel Oddi, An Uneasier Case for Copyright Than for Patent Protection of Computer Programs, 72 Nw. L. Rev. 351, 363 n.32 (1993).
\textsuperscript{220} AIPLA Report, supra note 209, at 63.
\textsuperscript{221} Id.
\textsuperscript{222} Id. at 64.
\textsuperscript{223} Id. at 71-72.
\textsuperscript{224} Id. at 64-66.
\textsuperscript{225} Harry T. Edwards, Alternative Dispute Resolution: Panacea or Anathema, 99 Harv. L. Rev. 668, 670 (1986).
\textsuperscript{226} 35 U.S.C. §§ 281, 283, 284.
for the infringement and are, at a minimum, a reasonable royalty.\textsuperscript{227} The court may triple the amount of damages\textsuperscript{228} and award attorney's fees.\textsuperscript{229}

In copyright cases, injunctions can also be obtained to prevent further infringement.\textsuperscript{230} The allegedly infringing goods can be impounded, including "all plates, molds, matrices, masters, tapes, film negatives, or other articles by means of which such copies may be reproduced."\textsuperscript{231} Actual damages or statutory damages can be recovered. Statutory damages are set by the court at $500 to $20,000 for any one work,\textsuperscript{232} and damages can be increased to $100,000 if the infringement was willful.\textsuperscript{233} Attorney's fees are available unless prohibited by statute.\textsuperscript{234}

Trade secret law varies from state to state. Therefore, the remedies vary from state to state as well. However, the Uniform Trade Secrets Act sets out the general guidelines.\textsuperscript{235} Actual or threatened misappropriation of trade secrets may be enjoined,\textsuperscript{236} and damages for actual loss caused by the misappropriation of the trade secret can also be recovered.\textsuperscript{237} Willful and malicious misappropriation can lead to a punitive damage award of not more than twice the actual damages.\textsuperscript{238} Attorney's fees can be recovered if the claim of misappropriation is made in bad faith, the motion to terminate an injunction is made or resisted in bad faith or the misappropriation was willful and malicious.\textsuperscript{239} In regard to the type of remedies available, there is not much difference between the various forms of protection. As a result, multiple types of protection do not provide any extra benefits.

The industry's view of the protection sought should also be taken into account. The software industry is slowly warming to the use of patents. Since the late 1970s, copyrights were the accepted form of protection. Prior to that time trade secrets were the accepted mode of

\textsuperscript{227} 35 U.S.C. § 284.
\textsuperscript{228} Id.
\textsuperscript{229} Id. 35 U.S.C. § 285.
\textsuperscript{230} 17 U.S.C. § 502.
\textsuperscript{231} Id. § 503.
\textsuperscript{232} Id. § 504(a). Note that statutory damages are only available if the copyright was registered before the commencement of the infringement or within three months of the first publication of the copyrightable work. Id. § 412.
\textsuperscript{233} Id. § 504.
\textsuperscript{234} Id. § 505: "Except as otherwise provided by this title, the court may also award a reasonable attorney's fees to the prevailing party as part of the costs."
\textsuperscript{235} See supra note 139 and accompanying text.
\textsuperscript{236} \textsc{Uniform Trade Secrets Act} § 2(a).
\textsuperscript{237} Id. § 3.
\textsuperscript{238} Id. § 3(b).
\textsuperscript{239} Id. § 4.
protection. There is still a large segment of the software industry that is generally adverse to patents. This may be because software programmers generally believe in the free flow of information or because the software industry itself is so young. Larger firms are now filing more software related patents, but many small producers of software and related goods still look upon patents with disapproval. The League for Programming Freedom, a group of programmers and software industry insiders, opposes almost all intellectual property protection for software. The Free Software Foundation also opposes software protection and specifically distributes free software. Such opposition has to be taken into account when considering what recommendation to make to a client.

How close the competitors are in terms of technological development has to be examined as well. The United States is a "first to invent" system of patent protection. However, neither copyright nor trade secrets protect against independent development. A trade secret can be lost if a second party develops and publicizes the same invention. Therefore, patent protection may be preferred if the competitors are very close together in technological development.

International considerations are also important. A discussion of international law with regards to software protection is outside the scope of this article. An excellent source for discussion of the international implications of each of these types of protection is Denis Karjala’s article on recent international developments.

These concrete factors listed above are a suggested starting point for the practitioner. The practitioner should also evaluate any specific factors that appear relevant in the context of the company or industry in question.

241. Id. at 1137-78 n.440.
243. For more information, contact the Free Software Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307.
244. See supra notes 118 and 144 and accompanying text.
245. See supra notes 162-170 and accompanying text.
VIII. CHECKLIST OF CONSIDERATIONS

This checklist will summarize the factors described above. It then notes whether each factor weighs in favor, against or is neutral on each type of protection.

<table>
<thead>
<tr>
<th>FACTOR</th>
<th>PATENT</th>
<th>COPYRIGHT</th>
<th>TRADE SECRET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access, Many people</td>
<td>Irrelevant</td>
<td>Irrelevant</td>
<td>Not recommended</td>
</tr>
<tr>
<td>Coexistence - Copyright</td>
<td>Good</td>
<td>N/A</td>
<td>Synergistic</td>
</tr>
<tr>
<td>Coexistence - Patent</td>
<td>N/A</td>
<td>Good</td>
<td>Weak</td>
</tr>
<tr>
<td>Coexistence - Trade Secret</td>
<td>Weak</td>
<td>Synergistic</td>
<td>N/A</td>
</tr>
<tr>
<td>Cost of Maintenance</td>
<td>Not recommended</td>
<td>Excellent</td>
<td>Good</td>
</tr>
<tr>
<td>Cost to Obtain</td>
<td>High</td>
<td>None</td>
<td>Low</td>
</tr>
<tr>
<td>Distribution, Internal</td>
<td>Not recommended</td>
<td>Good</td>
<td>Excellent</td>
</tr>
<tr>
<td>Distribution, Wide</td>
<td>Excellent</td>
<td>Good</td>
<td>Not recommended</td>
</tr>
<tr>
<td>Industry - Another will Patent</td>
<td>Excellent</td>
<td>Good</td>
<td>Not recommended</td>
</tr>
<tr>
<td>Industry, Close Competitors</td>
<td>Excellent</td>
<td>Good</td>
<td>Not recommended</td>
</tr>
<tr>
<td>Portfolio - Licensing Revenues</td>
<td>High</td>
<td>Low</td>
<td>Variable</td>
</tr>
<tr>
<td>Portfolio - Need Investors</td>
<td>Excellent</td>
<td>Good</td>
<td>Not recommended</td>
</tr>
<tr>
<td>Portfolio, will Cross-license</td>
<td>Excellent</td>
<td>Not recommended</td>
<td>Not recommended</td>
</tr>
<tr>
<td>Sales, without License</td>
<td>Excellent</td>
<td>Not recommended</td>
<td>Not recommended</td>
</tr>
<tr>
<td>SW - Ideas or Processes, New</td>
<td>Excellent</td>
<td>Not recommended</td>
<td>Good</td>
</tr>
<tr>
<td>SW - Level of Abstraction, High</td>
<td>Good</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>SW - No Effect on the Real World World</td>
<td>Not recommended</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>SW - Originality, Level Required</td>
<td>High</td>
<td>Low</td>
<td>Minimal</td>
</tr>
<tr>
<td>Team - Long term Needed</td>
<td>Good</td>
<td>Excellent</td>
<td>Good</td>
</tr>
<tr>
<td>Time pressure - currently infringed</td>
<td>Not recommended</td>
<td>Good</td>
<td>Excellent if misappropriated</td>
</tr>
<tr>
<td>Use, High Value</td>
<td>Excellent</td>
<td>Good</td>
<td>Not recommended</td>
</tr>
<tr>
<td>Use, Short Term</td>
<td>Not recommended</td>
<td>Excellent</td>
<td>Good</td>
</tr>
<tr>
<td>Willingness to enforce</td>
<td>Excellent</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Development is work-intensive, limited creativity</td>
<td>Not recommended</td>
<td>Good</td>
<td>Excellent</td>
</tr>
</tbody>
</table>

IX. CONCLUSION

Computer software is capable of being protected by a variety of means. Patent, copyright and trade secret all can protect software to various extents. Each of these protections has its strong and weak points. After accounting for the circumstances of the company for which one works, combining these forms of protections can forge the ideal package for any software creation.

Patent protection is the broadest, protecting the underlying processes and ideas as they are implemented in one of the statutory categories required by patent law. The software must meet rigorous standards in order to be granted a patent. Although patent protection only lasts for up to twenty years, the protection is comprehensive during those years. It protects against innocent infringers and independ-
ent creators, as well as all others who step within the area protected by the claims of the patent. Once the patent is issued, it is presumed to be valid and is generally upheld by the Court of Appeals for the Federal Circuit. Copyright, on the other hand, protects the expression used by the author, in this case the programmer, in creating the literary work that is a piece of software. The courts have extended this protection to cover nonliterary elements, such as the structures used by the program. But this extension is very limited and excludes such items as structure dictated by external circumstances or efficiency. Copyright protects only against the copying of the protected items. An independent creator, who happens upon the same structure, or even an almost identical program, is not liable. Yet copyright requires no effort on the part of the author or owner of the copyright, except fixation in a medium. Copyright protection endures for the life of the author plus fifty years, or if the work is made for hire, for seventy-five years from first publication or 100 years from date of creation.

Trade secret can protect every aspect of a program, from the underlying ideas to the actual expressions used by the programmer. In order for something to qualify as a trade secret, it must have a minimal amount of inventiveness, and its owner must take reasonable precautions to maintain secrecy. These precautions are generally easy, but in some situations they require extensive safeguards. Trade secret protection is only valuable as long as no one learns the secret through legitimate means. If that can be prevented, protection can last indefinitely.

Patent and copyright have an uneasy and, as of yet, untested cooperation. They have both been held to be valid forms of computer software protection, but there has not been a case to date that involved both types of protection. It appears, generally, that they are not mutually exclusive. A patented computer software program could also be the subject of copyright protection. In fact, since copyright attaches at the time the work is fixed, software is automatically protected.

Copyright and trade secret may coexist and enhance each other. By combining copyright and trade secret protection, the impact of reverse engineering, done in order to either eliminate the trade secret or find the uncopyrightable parts of the work, can be significantly decreased. Copyright Office regulations contain a specific provision for the inclusion of trade secrets in the disclosure.

Trade secret and patent have an uneasy alliance. In general, they cannot coexist. In limited situations, however, they can avoid cancelling each other out when the trade secret has known equivalents or when the patent is a small portion in a larger program protected by
trade secret. Even in such situations, these two forms of protection do not work together, but merely side by side.

The company being advised has to be taken into consideration as well. Its current status, as well as future goals, have to be evaluated to see how they fit in with each type of protection. Generally, if one type of protection appears overwhelmingly favored by the company, there is a reason for its preference. If two or more forms of protection actually appear to fit the situation in terms of preference and level of protection needed, using multiple protections may be appropriate.

The analogy between software and home protection seems appropriate. Just as everyone wishes to protect their home, every company is looking for the optimal form of protection for its software. Selecting the most appropriate type of software protection must be evaluated on a case-by-case basis. Using the checklist above, along with a thorough knowledge of a client and the client's objectives, the attorney can maximize protection of software.