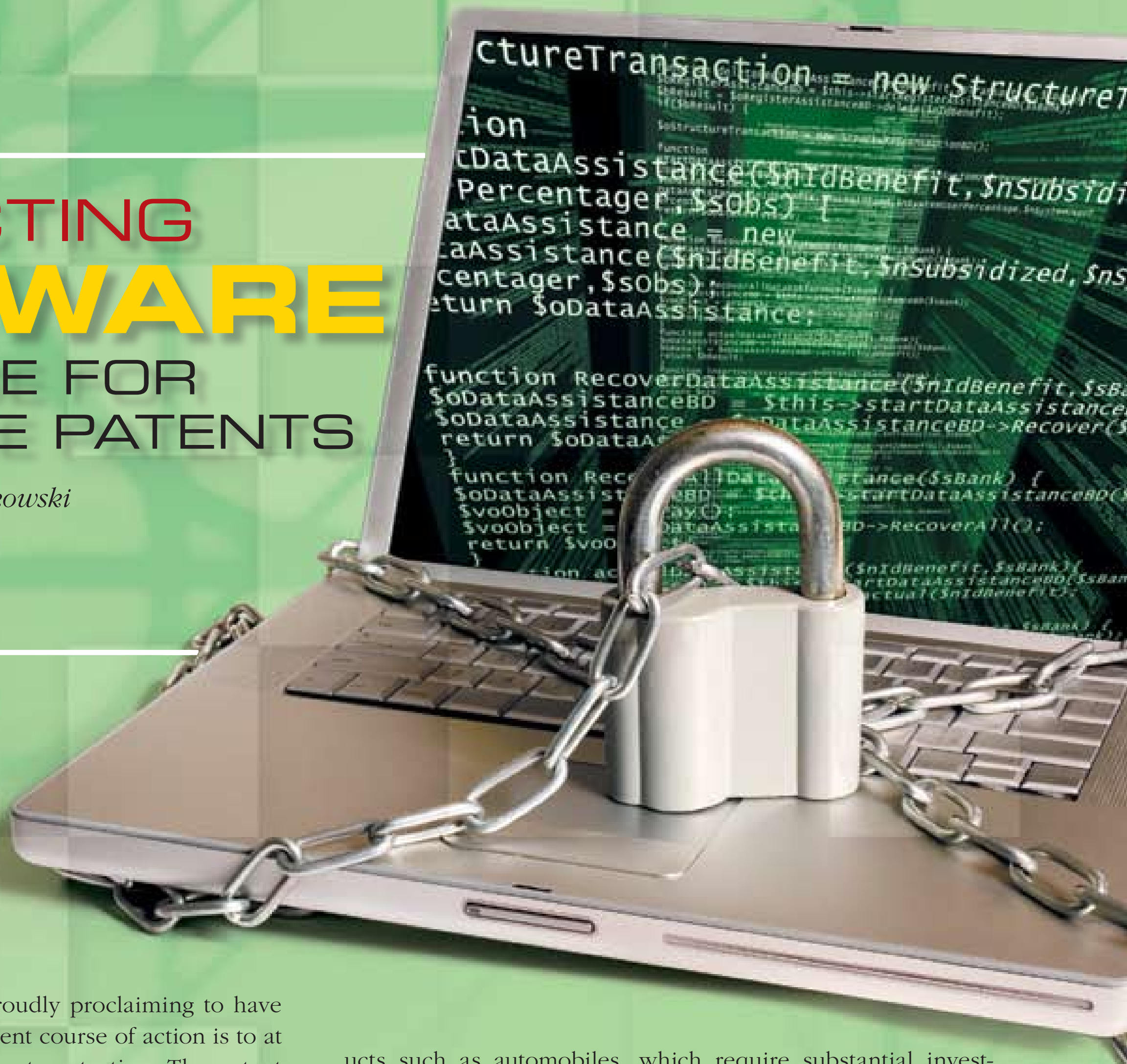
PROTECTING SOFTWARE

THE CASE FOR SOFTWARE PATENTS

By Chris Falkowski



INTRODUCTION

A client comes to your office, proudly proclaiming to have built "a better mousetrap." The prudent course of action is to at least consider the possibility of patent protection. The patent system was specifically created to provide inventors with limited monopolies over their "new and useful" inventions.[2] The advice to investigate patent protection is generally not controversial. However, the decision to pursue patent protection can become controversial when the "mousetrap" relates to computer software. The history of software patents, like the patent system generally, is not free from imperfections and flaws. In the past decade, there have been several prominent examples of software patents in the news of the day, and thus, software patents have been subject to substantial criticism, some of it well deserved. Many commentators focus on particular examples of software patents that they find egregious, while others exhibit skepticism or even hostility toward the eligibility of any software-related invention to be patented.

Regardless of the hype and hyperbole of the global debate on software patents, attorneys must zealously represent the interests of their clients. In the course of advising clients involved in the creation of software, it is important to evaluate the different alternatives available for protecting software. Unlike products such as automobiles, which require substantial investments in machines and material to produce, software can be mass produced and easily distributed on a worldwide basis to millions of users from a single copy of the software residing on a single computer with an Internet connection.

Software is more dependent on intellectual property protection than virtually any other type of product or service. In advising clients, it is important to fully evaluate the different alternatives for protecting software. In many contexts, patent protection for software inventions is like democracy in Winston Churchill's quote—a flawed option that is nonetheless superior to any alternative.

COPYRIGHT PROTECTION

COPYRIGHT LAW BASICS

Creators of software have traditionally looked to copyright law for protection. Copyright law is federal law under the U.S. Copyright Act,[3] a statute passed by Congress pursuant to Article I, Section 8 of the U.S. Constitution.[4] The U.S. Copyright Act provides a definition for what constitutes a "computer program."[5]

"[I]t has been said that democracy is the worst form of government, except for all the others that have been tried from time to time."

— Winston Churchill, 1947[1]

Software is subject to copyright protection as a "literary work."[6] Copyright protection attaches automatically to computer code or any other type of literary work as the work is being created. The only procedural requirement for copyright protection is the placement of a copyright notice in or on the software. Copyright protection does not require the registration of a copyright with the U.S. Copyright Office. However, the right to sue other parties in federal court for monetary damages does require that the plaintiff hold a federally registered copyright.[7] It is relatively easy and inexpensive to apply for a copyright registration using the information available on the Library of Congress website.[8]

IDEA/EXPRESSION DICHOTOMY

Copyright protection for software is easy and inexpensive to obtain. However, the scope of that protection is limited to the copying of creative expression and to copying the look and feel of the software. Copyright protection is limited by what is referred to as the "idea/expression" dichotomy.[9] Any factual or functional aspect of a copyrighted work is not and cannot be protected by copyright. "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 a work."[10]

"THIN" PROTECTION FOR SOFTWARE

As a result of the idea/expression dichotomy, software is protected by copyright law only to the extent that the software "possesses more than a de minimis quantum of creativity."[11] The creativity of a software program can reside in the source code or object code used to make the program run, or it can be attributed to the appearance and other aesthetic attributes of the software's interface with which the user interacts. Computer programs, even extremely function-oriented software such as operating systems, are likely to be found to possess the requisite level of creative expression to qualify for some measure of copyright protection.[12] However, such protection is generally considered to be "thin" protection.[13]

Thin protection is sufficient to prevent outright copying as evidenced by an infringing program that is identical to the copyrighted work. Thin protection also provides actionable protection when the infringing work is "virtually identical" to the copyrighted work, as was the case in *Apple Computer, Inc v Franklin Computer Corp.*[14] As long as the two works being compared are "identical" or "virtually identical," courts tend to conclude that there is a minimum level of creativity embodied in the original work to support a finding of infringement. However, when the differences between the copyrighted program and the allegedly infringing program are identified as being sufficiently non-trivial, the process for evaluating the scope of copyright protection will involve some type of "filtration" analysis in which alleged similarities between the two works are broken down and evaluated with respect to the idea/expression dichotomy.[15] Each individual element is evaluated with respect to whether it constitutes original expression or the implementation of an idea or function.

Even in disputes involving purely aesthetic literary works, such as screenplays or novels, this filtration process often represents an extremely difficult hurdle for the plaintiff to overcome. In the context of function-oriented software, the application of a filtration-based analysis is often fatal to the plaintiff's case as a practical matter unless there is evidence of verbatim copying.[16] At some point in the analysis, courts tend to conclude that there are only a relatively small number of ways to write software in an efficient manner and, as a result, that the idea

FAST FACTS:

The standards for patentability are no different for software inventions than they are for any other type of technology.

Software is more dependent on intellectual property protection than virtually any other type of product or service.

Copyright law protects a computer program to the extent that the program embodies creative expression, but it cannot protect the functionality or structure of the program.

IP OPTIONS—AT A GLANCE

	Copyrights	Trade Secrets	Trade Dress	Patents
Scope of Protection	• creative expression embodied in the program • "look and feel" of the software interface	• secret information that has economic value • can include algorithms performed by a computer program, design/architecture attributes, and even mere data stored on a database	• the "distinctiveness" of a computer program with respect to the perceptions of a user • similar to the "look and feel" protection under copyright law, except that it is evaluated from the framework of source identification and market perceptions	• functionality or structure or both
Advantages	• easy to obtain • inexpensive way to prevent outright copying	• easy to obtain • inexpensive • can protect data • often effective at protecting aspects invisible to the user, such as algorithms and design/architecture attributes	• easy to obtain • inexpensive • it is probably easier to show trade dress infringement than "look and feel" copyright infringement because a trade dress analysis focuses on the issues of source identification and the likelihood of confusion in the marketplace	• can protect functionality—unlike copyrights and trade dress • independent creation is *not* a defense • can be disclosed without forfeiture—unlike trade secrets
Disadvantages	• does not prevent others from copying the functionality of a computer program • value of most software is in functionality, not aesthetics • independent creation is a valid defense that invites competitors to use a "clean room" approach to develop competing products	• it is not always possible to use or market a trade secret while keeping it a secret • independent creation is a valid defense—must show misappropriation • administrative error can result in forfeiture of the asset	• limited to aspects of the software that are perceived by the user, such as appearance and sound	• expensive and time-consuming to obtain • discloses the invention to the public
Registration Process	optional	none	optional	mandatory
Duration	expires 70 years after the death of the author	potentially infinite	potentially infinite	expires 20 years from filing date
State/Federal	federal	state	federal and state	federal
Applicable Statute	U.S. Copyright Act 17 USC 101 et al.	Uniform Trade Secrets Protection Act MCL 445.1901 et al.	Lanham Act 15 USC 1051–1141(n)	U.S. Patent Act 35 USC 101 et al.

and expression embodied in the software have merged together, rendering the copied aspects of expression essentially unprotected under copyright law.

COPYRIGHT ASSESSMENT

The bottom line for protecting software under copyright law is that copyright law provides an effective way to prevent others from making identical or substantially identical copies of a client's software "mousetrap." However, savvy competitors can independently create the software's functionality by first defining the desirable functionality in the copyrighted software, and then creating competing software without reference to the copyrighted software. Development teams refer to this process as a "clean room" or "white room" approach. This process can be time-consuming and expensive in certain contexts, but has a high likelihood of success. Ironically, this means that the more innovative and valuable the software is, the less satisfying copyright law is to the software innovator because competitors will be willing to pay substantial sums of money to avoid the scope of another party's copyright protection if the end result will be something extremely valuable. In contrast, a less valuable end result may provide insufficient incentive for the process of independent creation.

TRADE DRESS PROTECTION

The scope of trade dress protection is similar to the "look and feel" protection provided under copyright law. Both apply only to how humans experience the software using their senses. Traditionally, humans interact with software through the senses of sight and sound, but technologies relating to smell and touch already exist. Both trade dress and copyright law can protect the appearance of the user interface on the display screen, including its windows, menus, graphic design, print sizes, and styles. The trade dress of a user interface can be protected to the extent that it is sufficiently distinctive to serve as a source identifier.

Trade dress law is a subset of trademark law, which exists at both the federal[17] and state levels. Trade dress protection has its underlying roots in state unfair competition law. While it is possible to register a trademark with the U.S. Patent and Trademark Office when the "mark" actually consists of the trade dress for a product or service,[18] such applications are often rejected on the basis of relating to functional aspects of the product or service.[19]

It may be somewhat easier to succeed in enforcing "look and feel" protection in the context of a trade dress or unfair competition claim than it is in the context of a copyright claim because the goal of trade dress law and unfair competition law is to prevent consumers from being confused about the origin of the goods or services. Unlike the test for copyright infringement, trade dress infringement, which applies a likelihood of confusion test, does not involve a filtration-based analysis that is very difficult to overcome in a software context. For this reason, some commentators wonder why Apple did not raise a trade dress claim against Microsoft in *Apple Computer, Inc v Microsoft Corp*,[20] the dispute over whether the Windows operating system infringed Apple's graphical user interface.

The bottom line with trade dress protection is that it does provide some level of protection for the appearance of the interface through which users interact with a computer program. Like copyright protection, trade dress protection is limited to non-functional elements. Trade dress does not protect functionality, and in most instances, functionality is what makes software useful.

TRADE SECRET PROTECTION

Trade secret law in Michigan is provided by the Uniform Trade Secrets Protection Act.[21] A trade secret under Michigan law is any "information" that "derives independent economic value, actual or potential, from not being generally known" and is subject to reasonable efforts "to maintain its secrecy."[22]

There is no process for registering trade secrets. The owner of a trade secret must take "reasonable" efforts to preserve the secrecy of the information. Thus, the time and cost of such efforts are dependent on the value of the trade secret. Coca-Cola must take significant efforts to preserve the secrecy of the formula for Coke, while a customer list for a barbershop is held to a significantly lower standard.

Trade secret protection is one of only two ways to potentially protect software's functionality. Unlike patent protection, trade secret can be immediate because it requires no registration process. In many contexts, trade secret protection is relatively inexpensive and consistent with prudent business practices, such as using nondisclosure agreements with vendors and customers.

The viability of trade secret protection is often determined by the critical question of whether it is possible to use a trade secret effectively while preserving its secrecy. Some software innovations are similar to the recipe for Coke—they can be used without giving away the secret. However, in many contexts, use of the software or even the marketing of the software's functionality inherently reveals the secret, negating the possibility of trade secret protection.

Even if it is possible to maintain a trade secret, trade secret protection is still deficient in one potentially important respect in contrast to patent protection. The enforcement of trade secret protection is limited to those who "misappropriate" the trade secret. Independent creation is a valid affirmative defense to trade secret misappropriation, but not to patent infringement. Virtually any trade secret dispute will involve a defendant claiming that her or she independently thought of the new "mousetrap." Many inventions are obvious in hindsight, so the obstacles faced by the plaintiff can be significant.

PATENT PROTECTION

The patent system provides inventors with the opportunity to obtain time-limited monopolies as long as the invention is

Patents can be time-consuming and expensive to obtain. The fact that a client could potentially obtain a patent does not mean that it makes business sense to do so.

new, useful, and non-obvious.[23] Many foreign jurisdictions have significant limitations on the types of software-related inventions that can be patented. The European Union's attempts to create a single unified patent system are being impeded by an inability to reach agreement on the treatment of software-related patents.[24] Although objections to software patents have not led to street protests in the U.S. as they have in Europe,[25] there are numerous examples of business, academic, and legal commentators voicing skepticism about software patents. Many of those criticisms are made specifically in the context of software patents, but the criticisms actually pertain to the patent system generally.

The patent system does not discriminate between different types of technologies.[26] Software patents are evaluated and issued under the same legal framework as other inventions such as pharmaceutical products, automotive components, consumer goods, medical devices, and a long litany of other technologies. Software is but one example of a technology eligible for patenting. The U.S. Supreme Court in *Diamond v Chakrabarty* proclaimed that "anything under the sun made by man"[27] can be patented with few exceptions. Despite the wide range of inventions that can be patented and despite the fact that patents involving software-related inventions are subject to the same rules and processes as other technologies, software patents are generally greeted with more skepticism and outright criticism than inventions such as piston rings, circuit boards, pharmaceutical products, pacemakers, and numerous other areas of technology.

CONCLUSION

Patents can be time-consuming and expensive to obtain. The fact that a client could potentially obtain a patent does not mean that it makes business sense to do so. However, in advising clients, it is important to understand that for many types of software inventions, patents provide the most meaningful protection available. Trade dress and copyright law cannot protect the functional aspects of a computer program, and it is typically functionality that makes software valuable. As a practical matter, copyright law and trade dress are often of limited value outside the context of verbatim copying. Trade secrets are often a viable option in the software industry with respect to embedded algorithms whose functionality cannot be precisely observed by a user of the software. However, it is impossible in many instances to use a trade secret effectively while still keeping it a secret, and trade secret litigation cannot preclude the invocation of an "independent creation" defense, a defense that is almost certain to be raised and likely difficult to overcome.

Given the limitations of the other available options, patent protection is an option that cannot be ignored. Whatever its flaws, the patent system is often the best way to protect the clients bearing better "mousetraps." That conclusion is true even if the particular "mousetrap" runs on a computer or is accessible using a web browser. ■

Chris Falkowski is a solo practitioner at Falkowski PLLC, focusing primarily on technology-related legal issues, including intellectual property and information technology matters. He is currently a council member and secretary for the Computer Law Section. Mr. Falkowski is a former editor of the Michigan Computer Lawyer*. Before attending law school, he worked as a software developer for a Fortune 150 company. Mr. Falkowski is a graduate of the engineering school at Northwestern University and the University of Michigan Law School.*

FOOTNOTES

[1] <http://thinkexist.com/quotation/it_has_been_said_that_democracy_is_the_worst_form/15815.html> (accessed May 5, 2007).

[2] 35 USC 101.

[3] 17 USC 101 et al.

[4] US Const, art I, § 8, clause 8 provides Congress with the power to "promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries."

[5] 17 USC 101.

[6] 17 USC 102(a)(1).

[7] 17 USC 411.

[8] See <http://www.copyright.gov> (accessed May 5, 2007).

[9] See *Feist Publications, Inc v Rural Telephone Service,* 299 US 340, 363 (1991).

[10] 17 USC 102(b).

[11] *Feist* at 363.

[12] See *Apple Computer, Inc v Franklin Computer Corp,* 714 F2d 1240 (CA 3, 1983) (infringement of operating system found when software was "virtually identical" and the "variations that did exist were minor, consisting merely of such things as deletion of reference to Apple or its copyright notice").

[13] See *Apple Computer, Inc v Microsoft Corp,* 35 F3d 1435 (CA 9, 1994).

[14] 714 F2d 1240 (CA 3, 1983).

[15] See *Computer Associates International, Inc v Altai, Inc,* 982 F2d 693 (CA 2, 1992).

[16] *Id.* (no infringement of program structure found after using the approach of "filtering out the unprotected aspects of an allegedly infringed program"); *Microsoft Corp, supra.*

[17] Lanham Act (15 USC 1051–1141(n)).

[18] Trademark Manual of Examination Procedures § 1202.02 (4th ed).

[19] Trademark Manual of Examination Procedures § 1202.02(a) (4th ed).

[20] *Microsoft Corp, supra.*

[21] MCL 445.1901 et al.

[22] MCL 445.1904.

[23] See 35 USC 101 and 103.

[24] See <http://news.com.com/2100-1012-998547.html> (accessed May 5, 2007).

[25] See <http://www.ecommercetimes.com/story/software/31431.html> (accessed May 5, 2007).

[26] See 35 USC 101 et al.

[27] *Diamond v Chakrabarty,* 447 US 303, 309 (1980).