Santa Clara High Technology Law Journal

Volume 14 | Issue 2 Article 4

January 1998

Reconciling Reverse Engineering and Conflicting Shrinkwrap License Terms Under U.C.C. Article 2B: A Patent Law Solution

Frank J. Pita

RECONCILING REVERSE ENGINEERING AND CONFLICTING SHRINKWRAP LICENSE TERMS UNDER U.C.C. ARTICLE 2B: A PATENT LAW SOLUTION[*]

Frank J. Pita[†]

I. INTRODUCING THE ISSUES

Proposed U.C.C. Article 2B seeks to standardize and legitimize software shrinkwrap licenses. "Shrinkwrap" is the common term for the transparent plastic or cellophane wrapping that seals boxes of mass marketed software. Software vendors typically attach written end user agreements to the shrinkwrap; hence, the term shrinkwrap licenses grew to represent the licenses themselves, even when these licenses appear on a computer screen instead of a box. The terms and conditions of the license become effective when the user rips open the package or otherwise manifests assent. Although these licenses are used pervasively to market software to consumers, they have been enforced inconsistently in the federal courts.[1]

[*] Copyright © 1998 Frank J. Pita.

[†] J.D. 1998, *Magna Cum Laude,* from North Carolina Central University School of Law. Mr. Pita has also earned B.S.E.E., M.S.E.E., and M.B.A. degrees from the University of Miami. He is registered to practice before the United States Patent and Trademark Office. Mr. Pita is currently an Associate Patent Attorney with the Bell Seltzer Intellectual Property Group of Alston & Bird in Raleigh, North Carolina. Previously, he worked for IBM Corporation in various computer and networking product design positions. His IBM design efforts generated two IBM patents and one Patent pending. While at IBM, Mr. Pita gained experience with reverse engineering by enforcing IBM hardware and software patents. The views expressed in this essay are solely those of the author, and not representative of the views of Alston & Bird, IBM Corporation, or North Carolina Central University. All errors and omissions remain those of the author. The author would like to thank Professor James Beckwith whose business insights and legal teachings helped inspire this essay, and Assistant Dean Adrienne Meddock, whose advice and encouragement helped this essay flourish.

[1] *See* the seminal case on the subject, ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1449 (7th Cir. 1996). For current commentary, *see* Carey R. Ramos & Joseph P. Verdon, *Shrinking and Click-On Licenses After ProCD v. Zeidenberg*, 13 COMPUTER/L.J. 1,1 (1996); Brandon L. Grusd, *Contracting Beyond Copyright: ProCD, Inc. v. Zeidenberg*, 10 HARV. J.L. & TECH. 353, 353 (1997).

Reverse engineering is "the process of starting with a finished product and working backwards to analyze how the product operates or how it was made."[2] Yet software shrinkwrap license terms typically prohibit reverse engineering, reverse compiling, or disassembly of software for any reason.[3] As a consequence, this creates a direct conflict with efforts to enforce software patents that usually require reverse engineering or other forbidden activities.

The law should allow software developers to protect their intellectual property. Developing software requires a significant investment in time and money and software retailers should be able to conditionally license their products freely under freedom of contract principles. However, this method of retailing software can frustrate the public's grant of a limited patent monopoly.

Protecting software as intellectual property has become critical as software use continues to permeate our society. However, the courts have not confronted many pivotal questions. Should federal law preempt contrary shrinkwrap license terms when reverse engineering is performed to uncover software patent infringement? How should limited patent monopolies be reconciled with freedom of contract? Should a reverse engineering right arise under patent law? How Congress and the courts address these questions will have enormous economic implications. As such, one possible answer to these questions is offered in this essay.

II. Software is Treasured in the New Economy

Technological advances are radically reshaping the world. Society must constantly struggle to evolve and keep up, lest it drown in a tidal wave of information. New profit opportunities emerge daily, while established markets go extinct just as quickly in our "hyper competitive" business climate.[4]

The software industry is the fastest growing sector of, and "a critical component, of the U.S. economy."[5] Why is this so? Software

[2] Secure Servs. Tech., Inc. v. Time & Space Processing, 722 F. Supp. 1354, 1361 (E.D. Va. 1989).

[3] David A. Rice, *Software License Prohibitions Against Reverse Engineering*, 53 U. PITT. L. REV. 543, 548 (1992).

[4] James B. Quinn et al., *Leveraging Intellect*, 10 ACAD. MGMT. EXEC., Aug. 1996, at 7-9.

[5] Public Hearings and Request for Comments on Patent Protection for Software-Related Inventions, 58 Fed. Reg. 66347, 66348 (Dec. 20, 1993).

is necessary for creating, exchanging, and managing the deluge of data flowing from this new economy. Also, massive revenues are at stake because modern society has grown to have an extreme dependence on software. Thus, the competition for profits is fierce among those innovators who recognize the new economic ground rules.

A stunning example: Bill, the Software Billionaire:

Bill Gates, founder of the Microsoft software empire, is again the world's richest person. One year ago, a respected publication estimated Bill Gates' personal net worth at more than $36.4 Billion.[6] *Forbes* magazine recently reported that Bill Gates' personal net worth has swelled to more than $51 Billion.[7] That company he works for is also doing quite well. Microsoft's stock capitalization is the fourth largest among all domestic U.S. corporations, and its cash balance is $9 Billion and multiplying.[8] This corporation has been on a spending spree as it acquired software competitors and media companies. Fearing that free market competition is being affected, the government has forced Microsoft to disgorge some acquisitions. Microsoft is so large that it has become a major Antitrust concern, warranting legal action by the federal government.

Presently, the Justice Department and almost half of the state attorney generals in the U.S. have filed antitrust claims against Microsoft. Allegedly, Microsoft's integration of its Internet browser into its Windows® 98 Operating System unfairly denies other software competitors' fair access to the desktop. Initial shipments of Windows 98 were delayed by an antitrust injunction. The Justice Department sought to fine Microsoft $1,000,000 per day for violating the 1995 consent decree that settled the antitrust suit. Bill and Microsoft have fiercely fought back and filed counterclaims against the federal and state governments. The ferocity of the counterattack reflects the incredible wealth of software treasure that Bill intends to protect. Those who recognize the new economic paradigms are shrewdly exploiting the massive riches that software presents.

[6] Kerry A. Dolan, *The Global Power Elite*, FORBES, July 28, 1997, at 98.

[7] Kerry A. Dolan, *The World's Working Rich*, FORBES, July 6, 1998, at 182; <http://www.forbes.com/forbes/98/0706/6201190a.htm>.

[8] John H. Christy, *The Forbes 500's: Market Value*, FORBES, April 21, 1997, at 212.

III. Protecting the Treasure

Software is the cornerstone of creative new products and technology, yet it is very easy to exploit. Several methods exist for protecting software intellectual property. Unfortunately, conflicts can arise among methods. Although typical shrinkwrap licenses prohibit reverse engineering, this is a primary means of uncovering software asset infringement. When shrinkwrap licensed software is reverse engineered to uncover intellectual property infringement, a protection conflict arises. The nature of software intellectual property protection may also vary according to the choice of methods used. Some methods of software protection are mutually exclusive, while others may be combined. The primary means of protecting software are Copyright, Patent, Trade Secret, and Contractual. For background, it is important to quickly review the relative merits of each method of protecting software assets.

Copyright

Copyright protection for software arrived in 1980, when Congress specifically amended § 101 of the Copyright Act[9] to protect computer programs, as evidenced by the inclusion of a definition of "computer program." A computer program is a "work of authorship" under the Copyright Act and thus worthy of protection.[10]

Copyright law protects the original expression of software ideas from being copied, but not any ideas, processes, or functions contained in the software.[11] Also, software must be fixed in some tangible media in order to be protected.[12] If the statutory requirements for copyright protection are satisfied, a software developer can copy, sell, modify, distribute, display, or perform the software creations.[13]

Obtaining a copyright registration is usually easy and inexpensive to obtain, however, copyright provides only limited protection for software as expression, without protecting its function.

[9] Copyright Act of 1976, 17 U.S.C. §§ 101-1101 (1994) ("Copyright Act"). Tandy Corporation v. Personal Micro Computers, Inc., 534 F. Supp. 171, 173 (N.D.Cal. 1981); *See* 2 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.08[A][2] (1996).

[10] 17 U.S.C. § 102(a). *See* Tandy Corporation v. Personal Micro Computers, Inc., 524 F. Supp. 171, 173.

[11] 17 U.S.C. § 102(a)-(b).

[12] 17 U.S.C. § 102(a).

[13] 17 U.S.C. § 106.

Patent

Patents are the most recent means of protecting software intellectual property. Patent law initially held that software was not patentable subject matter, but merely ideas or "mental steps."[14] Software was deemed an algorithm, not a process, and algorithms are not patentable subject matter.[15] A small exception allowing patentability eventually overcame the rule.[16] Software was held patentable subject matter as part of a patentable process, even if mathematical formulas or algorithms are used.[17] Software patents were allowed by examining the patentability of the process as a whole that incorporates computer software. Software executing in a general purpose computer was deemed patentable since it transformed the computer into a new special purpose machine.[18]

Driven by these favorable decisions, thousands of software related patents are processed today. Because of the growing volume of software inventions, the Patent Office has established specific examination guidelines.[19] Patent law can provide broad protection for software, if one can meet the stringent requirements for patentability: software programs must be useful, novel, and not obvious.[20] For a statutorily limited term, a patentee can exclude others from making, using, selling, or offering to sell the patented invention. However, obtaining a software patent can be very slow, difficult, and expensive.

Trade Secret

Trade secrets provide fragile but potentially eternal software protection for software, provided secrecy is maintained. Unlike federal patent and copyright protection, trade secrets are a creation of state law. A trade secret is any information, including computer programs, processes, or devices, that derives economic value from not being generally known to others.[21] Ideas and functions can sometimes be protected by trade secret.

[14] *In re* Abrams, 188 F.2d 165, 168 (C.C.P.A. 1951).

[15] Gottschalk v. Benson, 409 U.S. 63, 71 (1972).

[16] *Id.* at 77-78.

[17] *Id.* at 88.

[18] *In re* Alapatt, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc).

[19] Examination Guidelines for Computer-Related Inventions, U.S. Patent and Trademark Office, 61 Fed. Reg. 7478, February 28, 1996.

[20] 35 U.S.C. §§ 101-103 (1996).

[21] UNIF. TRADE SECRETS ACT § 1 (1985).

To preserve a trade secret, its holder must take reasonable steps to maintain secrecy.[22] Trade secrets terminate and become public domain information if publicly disclosed, for any reason.[23] Reverse engineering and independent discovery have been held legally viable means of ending another's trade secret, which of course impacts any trade secret material contained in software.[24] However, conditional or limited disclosure could preserve trade secret protection.

Trade secret protection may be quickly obtained, but can be destroyed just as quickly. Its costs are merely those of establishing and maintaining something as a secret. Trade secrets alone do not grant exclusive rights, while providing far weaker and more unstable protection than patents. Since inherent secrecy is hard to maintain for mass marketed software, trade secrets alone are usually too vulnerable to properly protect software.

Contract

Contract law, in the form of shrinkwrap license agreements, is the most popular method of protecting mass marketed software.[25] There are many advantages of software shrinkwrap licenses. First, ultimate control and copy prevention are important benefits of shrinkwrap licenses. Warranties can also be readily limited or excluded under a shrinkwrap license. Retailers gain transaction efficiency and contractual predictability from using boilerplate license terms. Compared with other forms of protecting software, license contracts are quick, easy, and inexpensive. However, license contracts alone cannot protect the functional aspects of software from independent competition.

The disadvantage for consumers and competitors is that software vendors combine trade secret and license agreement protections to create rights far beyond those under granted under patent and copyright law. Additionally, typical license agreements impose restraints on software use that are not freely bargained for by the consumer.

[22] UNIF. TRADE SECRETS ACT § 1 (1985). *See* The Gates Rubber Co. v. Bando Chemical Industries, Ltd., 9 F.3d 823, 848 (1993).

[23] Kewanee Oil, Co. v. Bicron Corp., 416 U.S. 470, 475-476 (1974).

[24] *Id.* at 490.

[25] Mark A. Lemley, *Intellectual Property and Shrinkwrap Licenses,* 68 S. CAL. L.REV., 1239, 1246.

III. The Law Adapts to the New Digital Economy

The Uniform Commercial Code ("U.C.C.") reflects the era during which it was drafted — a 1950's pre-computer-age economic model. Then, clear lines existed between goods, intangible assets, and services. As fundamental contract law governing tangible assets, Article 2 of the U.C.C. presumes a sale or other transaction which transfers title and possession; it is optimized for bargains in tangible goods wherein inherent physical characteristics provide value.

However, the service and information sectors of modern economies are larger than most manufacturing sectors. Thus, today's economies value information and services more than traditional tangible goods. Article 2 needed revision, because it failed to properly address transfers of modern intangible goods, such as software or digital information. In addition, the typical digital retail transaction is a conditional license of software that does not transfer title.

Since most software that is retailed today is licensed, not sold, a typical shrinkwrap license for mass marketed software is neither a sale nor a lease.[26] Software license transactions typically grant a conditional right to use software without passing title, so the only tangible item that a customer, or "end user," might purchase is any tangible media containing the software.[27]

Structuring a software transaction this way provides several advantages. Since software retailers maintain ownership of licensed software, these licenses are used to prohibit customers from making unauthorized copies of licensed software for use or resale. Licenses can limit how software may be used, with one important limitation being the one against reverse engineering: prohibitions against disassembly, decompilation and other reverse engineering methods are standard terms found in these licenses.

Yet, shrinkwrap licenses are unlike typical contracts where the parties meet, bargain and then reach mutual agreement. The software retailer delivers the licensed product with its attached terms and conditions; a customer must manifest assent through some affirmative act.[28] Assent by conduct could occur by opening a seal or shrinkwrap plastic package, clicking a button on a window, or entering some

[26] U.C.C. 2B, Basic Themes, Licensing Law and Practice (July 24, 1998 draft).

[27] D.C. Toedt III, *Shrinkwrap License Enforceability Issues,* 35 PRAC. L. 613, 617 (1996).

[28] U.C.C. 2B §§ 2B-203, 2B-112 (July 24, 1998 draft).

password.[29] Since a customer cannot modify or select among the terms, however, the model is truly a "take it or leave it" proposition for the consumer.

Although judicial enforcement of these licenses has been erratic, Article 2B will lay the statutory foundation so that software developers can presume that shrinkwrap licenses will be enforced as written. Indeed, Article 2B is being drafted specifically to address the new markets in intangible goods, such as the purchase of software.

IV. Shrinkwrap License Enforceability

Yet, the "adhesion contract" nature of these licenses threatens their enforceability, as their terms have been viewed as unbargained modifications to established contracts. An adhesion contract is a bargain drafted unilaterally by the dominant party, and presented as a final offer to a weaker party.[30] The terms are presented as a preprinted form to the weaker party, who lacks any realistic ability to negotiate the terms. An adhesion contract is generally not enforced since it is inequitable and not a valid bargain — violating both the letter and spirit of fundamental contract law.

Draft Article 2B of the U.C.C., *presumes enforceable licenses* so long as an end user has a right to a refund.[31] The common law contractual logic that supports enforcement of software licenses is evident throughout Article 2B.[32] The retailer is master of the offer, who can direct acceptance by conduct or any reasonable method. License terms, even if boilerplate forms, are enforceable if assent occurs by conduct.[33] Licenses are not deemed adhesion contracts if a consumer has the right to return software for a refund, and any unconscionable terms will not be enforced.[34] License terms are preempted and unenforceable to the extent they conflict with federal law.[35]

[29] U.C.C. 2B Part 2 Standard Forms and Manifesting Assent (July 24, 1998 draft).

[30] Vault Corp. v. Quaid Software Ltd., 655 F. Supp. 750, 760 (E.D. La. 1987), *aff'd*, 847 F.2d 255 (5th Cir. 1988).

[31] U.C.C. 2B §§ 2B-102(25), 2B-105, 2B-107, 2B-111, 2B-112, 2B-303, (July 24, 1998 draft).

[32] U.C.C. 2B, Part 2: Basic Themes, Freedom of Contract (July 24, 1998 draft).

[33] 2B § 303.

[34] 2B § 111.

[35] U.C.C. 2B, Part 2: Basic Themes, Intellectual Property Overlay (July 24, 1998 draft).

Despite intellectual property concerns, Article 2B will probably serve to legitimize and entrench these mass market contracts known as shrinkwrap licenses.

V. Violating Shrinkwraps by Reverse Engineering

The proposed statutory language in U.C.C. Article 2B does not directly address reverse engineering. Article 2B addresses the customary prohibition of reverse engineering as merely another limitation of rights within a shrinkwrap license. Specific terms in software licenses are not specifically addressed since Article 2B chooses to only provide "a generic contract law framework."[36]

Reverse engineering is recognized in Article 2B as a controversial matter.[37] Article 2B does recognize the possibility of preemption by federal intellectual property law. Despite Article 2B's detachment, shrinkwrap license terms universally prohibit reverse engineering, decompiling, or disassembling software for any reason. These licenses also typically restrict the number of copies of software that can be made by a consumer.

Copying software into a target computer's memory is required to reverse engineer software. This has been deemed to make a copy under copyright law. Breach of license occurs when copying and reverse engineering shrinkwrap licensed software under Article 2B. Section 2B-708 indicates that breach of a shrinkwrap license results by violating its terms once assent is manifested. Sections 2B-707 and 2B-708 address the potential damages and remedies once a license has been breached by a software end user. Thus, by reverse engineering shrinkwrap licensed software, an end user breaches the conditional license on at least two grounds. A serious conflict develops when reverse engineering software occurs for patent enforcement purposes. Patentees intending to protect their intellectual property by reverse engineering face liability for violating a shrinkwrap license under Article 2B. This is an undesirable and unintentional conflict between alternative mechanisms for protecting software intellectual assets.

[36] U.C.C. 2B at part 2: Intellectual Property Overlay, at 16 (July 24, 1998 draft).

[37] U.C.C. 2B § 2B-105, Reporter's Notes, at 63 (July 24, 1998 draft).

VI. WHERE DO WE GO FROM HERE?

In any conflict, it is important to consider the relative merits and motivations of competing interests. Given the design of Article 2B, it is clear that shrinkwrap license terms will be enforced and upheld. Since Article 2B defers to federal intellectual property law, reasonable limitations on freedom of contract exist. Federal preemptive power should permit good faith reverse engineering to uncover patent infringement in this context. Antitrust considerations also exist.

The law should permit software intellectual property to be protected, whether by patent or trade secret. Software today is being protected by a hybrid mechanism. The combination of trade secret protection and shrinkwrap licenses prohibiting reverse engineering is a powerful mechanism. However, patent monopolies granted by the public should prevail over any private monopolies created by contracts under state law.

Software vendors should not be allowed to abuse contract law. Propagating indefinite restraints on trade without benefiting the public is wrong. Contracts should not create rights beyond what federal laws and the public grant. Benefits from these shrinkwrap licenses flow primarily to the software vendor. The practical effect of restraints on reverse engineering are to encourage secret invention that will never benefit the public. Patent systems should be favored since granting patents makes new creations and technology generally available to the public. Trade secrets protected by license restrictions may never benefit anyone beyond the owner.

A shrinkwrap license term should be a shield rather than a sword. It should not be used as a sword to attack patentee rights and to dissect patent monopolies. Contract law should not be used to frustrate patent monopolies granted by the public. Rather, a shrinkwrap license should be enforced as a shield against unlawful violations of copyright or trade secrets.

Above all, fairness and encouraging innovation to benefit the public should remain as the guiding principles in this dispute. Burdens as well as benefits need to be balanced in a viable solution.

VI. RECOMMENDATION

Federal patent law should codify a right to reverse engineer as a good faith means of enforcing a valid patent. Legal exposure and harm resulting form reverse engineering can then be eliminated. The scope of the right to reverse engineer can be clearly defined, uncertainty and conflict can be reduced, and protection for all types of software intellectual assets can be maximized. Equity and balance can be achieved among the alternative mechanisms for protecting software intellectual property. These mechanisms can then be used to complement each other instead of conflicting.

By analogy, federal copyright law has already established a fair use exception allowing reverse engineering for the purpose of copyright enforcement.[38] Surely reverse engineering should be allowed when intended to enforce a legitimate property right. Contract law was never intended as a sword for carving away patent rights granted by the public.

New U.C.C. Article 2B will further propagate software shrinkwrap licenses. By its terms, Article 2B acknowledges its limitations and defers to federal intellectual property law. Therefore, Article 2B will intersect well with a new federal statutory right to reverse engineer for patent enforcement. This is a reasonable way to harmonize shrinkwrap license protection rights with patent property rights.

This recommendation provides a patent law solution that reconciles contract law and patent law. Hopefully, the patent statute will be amended before the conflict is realized. May the drafting begin!

[38] 17 U.S.C. § 107.